



Guia do usuário da versão 1

AWS Command Line Interface



AWS Command Line Interface: Guia do usuário da versão 1

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

.....	xiii
Sobre o AWS CLI	1
Sobre a AWS CLI versão 1	2
Manutenção e suporte para as versões SDK principais	2
Sobre a Amazon Web Services	2
Sobre os exemplos	3
Documentação e recursos adicionais	4
AWS CLI documentação e recursos	4
Outros AWS SDKs e ferramentas	4
Instalar a AWS CLI	6
Requisitos da versão do Python	6
Amazon Linux	7
Pré-requisitos	7
pip	8
yum	9
Solução de problemas de erros de AWS CLI instalação e desinstalação	10
Linux	10
Pré-requisitos	10
Instalar e desinstalar usando o instalador empacotado	11
Instalar e desinstalar usando pip	17
Instalar e desinstalar usando pip	20
Adicione o executável da AWS CLI versão 1 ao seu caminho de linha de comando	22
Solução de problemas de erros de AWS CLI instalação e desinstalação	23
macOS	23
Pré-requisitos	24
Instalar e desinstalar usando o instalador empacotado	24
Instalar e atualizar usando pip	30
Solução de problemas de erros de AWS CLI instalação e desinstalação	34
Windows	34
Instale, atualize e desinstale usando o MSI instalador	34
Instalar, atualizar e desinstalar usando Python e pip	36
Adicione o AWS CLI executável ao seu caminho de linha de comando	38
Solução de problemas de erros de AWS CLI instalação e desinstalação	40
Virtualenv	40

Pré-requisitos	40
Instalar e atualizar em um ambiente virtual	41
Solução de problemas de erros de AWS CLI instalação e desinstalação	42
Configurar o AWS CLI	43
Precedência de credenciais e configurações	43
Tópicos adicionais nesta seção	44
Configurações do arquivo de configuração e credenciais	44
Formato dos arquivos de credenciais e configuração	45
Onde as definições de configuração ficam armazenadas?	50
Usar perfis nomeados	51
Definir e visualizar as configurações usando comandos	52
Definir novos exemplos de comandos de configuração e credenciais	54
Configurações de arquivo config compatíveis	55
Variáveis de ambiente	73
Como definir variáveis de ambiente	74
AWS CLI variáveis de ambiente suportadas	75
Opções de linha de comando	85
Como usar as opções de linha de comando	86
AWS CLI opções de linha de comando global suportadas	86
Usos comuns das opções de linha de comando	90
Conclusão de comando	90
Como funciona	91
Configuração do preenchimento automático de comando no Linux ou no macOS	92
Configuração do preenchimento de comandos no Windows	95
Repetições	97
Modos de novas tentativas disponíveis	97
Configuração um modo de nova tentativa	100
Visualização de logs de novas tentativas	101
Use um HTTP proxy	102
Como usar os exemplos da	102
Autenticar para um proxy	103
Usando um proxy em EC2 instâncias da Amazon	104
Solução de problemas	104
Endpoints	105
Definir o endpoint para um único comando	105
Defina um endpoint global para todos Serviços da AWS	105

Definido para usar FIPs endpoints para todos Serviços da AWS	107
Configurado para usar os endpoints de pilha dupla para todos os Serviços da AWS	108
Definir endpoints específicos de serviço	109
Precedência de configurações e definições do endpoint	113
Autenticação e credenciais de acesso	114
Precedência de credenciais e configurações	114
Tópicos adicionais nesta seção	115
Credenciais de curto prazo	115
IAMfunções	117
Pré-requisitos	117
Visão geral do uso de IAM funções	117
Configurar e usar uma função	119
Usando MFA	121
Funções entre contas e ID externo	122
Especificar um nome de sessão de função para facilitar a auditoria	123
Assumir a função com a identidade da web	124
Limpar as credenciais em cache	125
IAMusuários	126
Etapa 1: Crie seu IAM usuário	126
Etapa 2: obter as chaves de acesso	126
Configure o AWS CLI	127
Use metadados de EC2 instâncias da Amazon para credenciais AWS CLI	128
Pré-requisitos	128
Configurando um perfil para metadados da Amazon EC2	129
Credenciais externas	130
Use o AWS CLI	133
Obter ajuda	134
O comando de ajuda integrado da AWS CLI	134
AWS CLI guia de referência	139
APIdocumentação	139
Solucionar de problemas de erros	140
Ajuda adicional	140
Estrutura do comando	140
Estrutura do comando	140
Comandos de espera	142
Especificar valores de parâmetro	143

Tipos comuns de parâmetros	144
Aspas com strings	149
Parâmetros de arquivos	153
Gerar um modelo de esqueleto da CLI	157
Sintaxe Simplificada	163
Controlar a saída do comando	166
Saída sensível	166
Opções de saída do lado do servidor e do lado do cliente	167
Output Format	168
Paginação	175
Filtragem da saída da	178
Códigos de retorno	202
Aliases	203
Pré-requisitos	204
Etapa 1: Criação do arquivo de alias	204
Etapa 2: Criação de um alias	205
Passo 3: Como chamar um alias	208
Exemplos de repositório de alias	210
Recursos	212
Exemplos de código	213
Exemplos de comando guiado	213
DynamoDB	214
Amazon EC2	218
S3 Glacier	238
IAM	245
Amazon S3	249
Amazon SNS	268
Exemplos de comando	270
ACM	277
APIGateway	288
APIGateway HTTP e WebSocket API	345
APIGerenciamento de gateway API	391
App Mesh	393
App Runner	438
AWS AppConfig	473
Application Auto Scaling	508

Application Discovery Service	525
AppRegistry	532
Athena	544
Auto Scaling	578
Planos de Auto Scaling	649
AWS Backup	657
AWS Batch	663
AWS Budgets	678
Amazon Chime	688
Controle de nuvem API	762
AWS Cloud Map	768
AWS Cloud9	778
AWS CloudFormation	787
CloudFront	837
Amazon CloudSearch	906
CloudTrail	907
CloudWatch	924
CloudWatch Registros	939
CloudWatch Monitoramento de rede	945
CodeArtifact	957
CodeBuild	986
CodeCommit	1051
CodeDeploy	1126
CodeGuru Avaliador	1167
CodePipeline	1186
AWS CodeStar	1218
AWS CodeStar Notificações	1232
CodeConnections	1243
Identidade do Amazon Cognito	1251
Provedor de identidade do Amazon Cognito	1257
Amazon Comprehend	1325
Amazon Comprehend Medical	1463
AWS Config	1499
Amazon Connect	1523
AWS Cost and Usage Report	1540
Serviço Cost Explorer	1543

Firehose	1551
Amazon Data Lifecycle Manager	1554
AWS Data Pipeline	1561
DataSync	1570
DAX	1574
Detecção	1594
Device Farm	1605
AWS Direct Connect	1610
AWS Directory Service	1662
AWS DMS	1665
Amazon DocumentDB	1709
DynamoDB	1767
DynamoDB Streams	1864
Amazon EC2	1871
Amazon EC2 Instance Connect	2545
Amazon ECR	2546
Amazon ECS	2577
Amazon EFS	2664
Amazon EKS	2672
Elastic Beanstalk	2751
Elastic Load Balancing - Versão 1	2782
Elastic Load Balancing - Versão 2	2810
Elastic Transcoder	2864
ElastiCache	2891
MediaStore	2998
Amazon EMR	3015
Amazon EMR em EKS	3065
EventBridge	3067
Firewall Manager	3073
AWS FIS	3083
Amazon GameLift	3102
Global Accelerator	3136
AWS Glue	3175
GuardDuty	3197
AWS Health	3215
HealthImaging	3223

HealthLake	3246
HealthOmics	3258
IAM	3327
Analizador de acesso do IAM	3467
Image Builder	3502
Incident Manager	3544
Incident Manager Contacts	3567
Amazon Inspector	3590
AWS IoT	3621
AWS IoT 1-Click Dispositivos	3805
AWS IoT 1-Click Projetos	3815
AWS IoT Analytics	3826
Device Advisor	3854
AWS IoT data	3869
AWS IoT Events	3872
AWS IoT Events-Data	3897
AWS IoT Greengrass	3922
AWS IoT Greengrass V2	4010
AWS IoT Jobs SDK release	4036
AWS IoT SiteWise	4040
AWS IoT Things Graph	4090
AWS IoT Wireless	4118
Amazon IVS	4155
Amazon IVS Chat	4194
Streaming IVS em tempo real da Amazon	4208
Amazon Kendra	4234
Kinesis	4243
AWS KMS	4262
Lake Formation	4332
Lambda	4384
License Manager	4427
Lightsail	4440
Macie	4566
Amazon Managed Grafana	4572
MediaConnect	4573
MediaConvert	4590

MediaLive	4614
MediaPackage	4621
MediaPackage VOD	4636
MediaStore Plano de dados	4648
MediaTailor	4654
MemoryDB	4659
Amazon MSK	4696
Gerenciador de rede	4705
Nimble Studio	4743
OpenSearch Serviço	4762
AWS OpsWorks	4776
AWS OpsWorks CM	4834
Organizações	4849
AWS Outposts	4888
AWS Payment Cryptography	4892
AWS Payment Cryptography Plano de dados	4913
Amazon Pinpoint	4922
Amazon Polly	4946
AWS Price List	4952
AWS Private CA	4957
AWS Proton	4965
QLDB	4977
Amazon RDS	5001
Amazon RDS Data Service	5203
Amazon RDS Performance Insights	5207
Amazon Redshift	5211
Amazon Rekognition	5293
AWS RAM	5370
Explorador de recursos	5394
Resource Groups (Grupos de recursos)	5416
Marcação de Resource Groups API	5430
AWS RoboMaker	5434
route 53	5470
Registro de domínios do Route 53	5484
Route 53 Resolver	5512
Amazon S3	5557

Amazon S3 Control	5650
S3 Glacier	5666
Secrets Manager	5688
Security Hub	5717
AWS Serverless Application Repository	5797
Service Catalog	5799
Service Quotas	5831
Amazon SES	5841
Shield (Escudo)	5854
Signer	5871
Snowball	5880
Amazon SNS	5882
Amazon SQS	5904
Storage Gateway	5925
AWS STS	5928
AWS Support	5937
Amazon SWF	5950
Systems Manager	5967
Amazon Textract	6145
Amazon Transcribe	6156
Amazon Translate	6200
Trusted Advisor	6202
Permissões verificadas	6222
VPCTreliça	6249
AWS WAF Classic	6277
AWS WAF Classic regional	6282
AWS WAFV2	6289
Amazon WorkDocs	6334
Amazon WorkMail	6367
Fluxo de WorkMail mensagens da Amazon	6391
WorkSpaces	6393
X-Ray	6409
Exemplos de script Bash	6426
DynamoDB	6427
Amazon EC2	6499
HealthImaging	6605

IAM	6614
Amazon S3	6669
AWS STS	6693
Segurança	6696
Proteção de dados	6697
Criptografia de dados	6698
Identity and Access Management	6698
Público	6699
Autenticando com identidades	6699
Gerenciando acesso usando políticas	6703
Como Serviços da AWS trabalhar com IAM	6706
Solução de problemas AWS de identidade e acesso	6706
Compliance Validation	6708
Resiliência	6709
Segurança da infraestrutura	6710
Aplicando uma versão mínima TLS	6710
Solucionar erros	6715
Solução geral de problemas para tentar primeiro	6715
Verifique a formatação AWS CLI do seu comando	6716
Verifique o que Região da AWS seu AWS CLI comando está usando	6716
Confirme se você está executando uma versão recente da AWS CLI	6717
Como usar a opção --debug	6717
Ative e revise os registros do histórico de AWS CLI comandos	6723
Confirme se o seu AWS CLI está configurado	6723
Erros de comando não encontrado	6724
O comando "aws --version" retorna uma versão diferente da que você instalou	6727
O comando "aws --version" retorna uma versão após a desinstalação do AWS CLI	6728
O comando AWS CLI processado com um nome de parâmetro incompleto	6729
Erros de acesso negado	6730
Credenciais inválidas e erros de chave	6731
Assinatura não corresponde aos erros	6733
Erros de nenhum console do Windows encontrado	6734
SSLerros de certificado	6735
Erros inválidos JSON	6736
Recursos adicionais	6738
Histórico do documento	6739

Esta documentação é para a AWS CLI única versão 1. Para obter a documentação relacionada à versão 2 do AWS CLI, consulte o [Guia do usuário da versão 2](#).

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

O que é o AWS Command Line Interface versão 1?

Note

A AWS CLI versão 1 não é a versão mais recente do AWS CLI. Alguns recursos apresentados na AWS CLI versão 2 não são compatíveis com a versão 1, e você deve fazer a atualização para acessá-los. Há algumas alterações "radicais" em relação à versão 1 que podem exigir alterações nos scripts. Para obter uma lista das alterações mais recentes na versão 2, consulte [Alterações mais recentes](#) no Guia do usuário da AWS CLI versão 2.

O AWS Command Line Interface (AWS CLI) é uma ferramenta de código aberto que permite que você interaja com AWS serviços usando comandos em seu shell de linha de comando. Com configuração mínima, o AWS CLI permite que você comece a executar comandos que implementam funcionalidades equivalentes às fornecidas pelo navegador baseado no prompt AWS Management Console de comando em seu programa de terminal:

- Shells do Linux: use programas comuns de shell, como [bash](#), [zsh](#) e [tcsh](#) para executar comandos no Linux ou macOS.
- Linha de comando do Windows — No Windows, execute comandos no prompt de comando do Windows ou em PowerShell.
- Remotamente — Execute comandos em instâncias do Amazon Elastic Compute Cloud EC2 (Amazon) por meio de um programa de terminal remotoSSH, como PuTTY ou `ssh`. AWS Systems Manager

Todas as funções de AWS administração, gerenciamento e acesso de IaaS (infraestrutura como serviço) no AWS Management Console estão disponíveis no e. AWS API AWS CLI AWS Os novos recursos e serviços de IaaS fornecem AWS Management Console funcionalidade completa durante API e no lançamento ou CLI em até 180 dias após o lançamento.

AWS CLI Fornece acesso direto ao público APIs de AWS serviços. Você pode explorar os recursos de um serviço com o AWS CLI e desenvolver scripts de shell para gerenciar seus recursos. Além dos comandos API equivalentes de baixo nível, vários AWS serviços fornecem personalizações para o. AWS CLI As personalizações podem incluir comandos de alto nível que simplificam o uso de um serviço com um complexo. API

Sobre a AWS CLI versão 1

A AWS CLI versão 1 é a original AWS CLI e continuamos a apoiá-la. No entanto, os principais novos recursos introduzidos na AWS CLI versão 2 podem não ser transferidos para a AWS CLI versão 1. Para usar esses recursos, você deve instalar a AWS CLI versão 2. A AWS CLI versão 1 foi criada usando o SDK for Python e, portanto, requer a instalação de uma versão compatível do Python.

Para instalar a AWS CLI versão 1, consulte [Instalar a AWS CLI](#).

Para verificar a versão atualmente instalada, use o seguinte 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
```

Para o histórico de versões, consulte o [Changelog da AWS CLI versão 1](#) ativado. GitHub

Manutenção e suporte para as versões SDK principais

Para obter informações sobre manutenção e suporte para as versões SDK principais e suas dependências subjacentes, consulte o seguinte no [Guia de referência de ferramentas AWS SDKs e ferramentas](#):

- [AWS SDKse política de manutenção de ferramentas](#)
- [AWS SDKsmatriz de suporte de versões e ferramentas](#)

Sobre a Amazon Web Services

A Amazon Web Services (AWS) é um conjunto de serviços de infraestrutura digital que os desenvolvedores podem utilizar ao desenvolver suas aplicações. Os serviços incluem computação, armazenamento, banco de dados e sincronização de aplicativos (mensagens e filas). AWS usa um modelo pay-as-you-go de serviço. Você será cobrado apenas pelos serviços que você ou suas aplicações usam. Além disso, para torná-la AWS mais acessível como plataforma para prototipagem e experimentação, AWS oferece um nível de uso gratuito. Neste nível, os serviços são gratuitos abaixo de um determinado nível de uso. Para obter mais informações sobre AWS custos e o nível gratuito, consulte [Nível AWS gratuito](#). Para obter uma AWS conta, abra a [página AWS inicial](#) e escolha Criar uma AWS conta.

Sobre os AWS CLI exemplos

Os exemplos AWS Command Line Interface (AWS CLI) neste guia são formatados usando as seguintes convenções:

- Prompt: o prompt de comando usa o prompt do Linux e é exibido como (`$`). Para comandos específicos do Windows, `C:\>` é usado como prompt. Não inclua prompt quando você digitar comandos.
- Diretório: quando comandos devem ser executados de um diretório específico, o nome do diretório é mostrado antes do símbolo do comando.
- Entrada do usuário o texto do comando inserido na linha de comando é formatado como **user input**.
- Texto substituível — O texto variável, incluindo nomes de recursos escolhidos por você ou IDs gerados por AWS serviços que você deve incluir nos comandos, é formatado como *replaceable text*. Em comandos de várias linhas ou comandos em que é necessária uma entrada de teclado específica, os comandos do teclado também podem ser mostrados como texto substituível.
- Saída — A saída retornada pelos AWS serviços é mostrada na entrada do usuário e é formatada como `computer output`.

O exemplo de comando **aws configure** a seguir inclui entradas do usuário, texto substituível e saída:

1. Insira **aws configure** na linha de comando e pressione Enter.
2. A AWS CLI saída de linhas de texto solicitando que você insira informações adicionais.
3. Insira cada uma de suas chaves de acesso e pressione Enter.
4. Em seguida, insira um nome de AWS região no formato mostrado, pressione Enter e, em seguida, pressione Enter uma última vez para pular a configuração do formato de saída.
5. O comando final Enter é mostrado como texto substituível porque não há entradas do usuário para essa linha.

```
$ 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
```


O exemplo a seguir mostra um comando simples com saída. Para usar este exemplo, insira o texto completo do comando (o texto destacado após o prompt) e pressione Enter. O nome do grupo de segurança, *meu-sg*, é substituível pelo nome do grupo de segurança desejado. O JSON documento, incluindo os colchetes, é gerado. Se você configurar sua CLI saída em formato de texto ou tabela, a saída será formatada de forma diferente. [JSON](#) é o formato de saída padrão.

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

Documentação e recursos adicionais

AWS CLI documentação e recursos

Além deste guia do usuário, a seguir estão recursos on-line valiosos quando você usa AWS CLI o.

- [AWS CLI guia de referência da versão 1](#)
- [AWS CLI Repositório de exemplos de códigos de script](#) Bash. Exemplos de scripts bash de código aberto. Os exemplos de scripts do Bash estão hospedados no [Repositório de exemplos AWS de código](#) em. GitHub
- [AWS CLI GitHub repositório](#). Você pode visualizar e bifurcar o código-fonte do AWS CLI on GitHub. Junte-se à comunidade de usuários GitHub para fornecer feedback, solicitar recursos e enviar suas próprias contribuições. Isso inclui visualizar e fornecer exemplos de comandos para a AWS CLI documentação.
- [AWS CLI repositório de exemplos de aliases](#) Você pode visualizar e bifurcar exemplos de AWS CLI aliases em. GitHub
- [AWS CLI Changelog da versão 1](#)
- [AWS CLI Changelog da versão 2](#)

Outros AWS SDKs e ferramentas

Dependendo do seu caso de uso, talvez você queira escolher uma das ferramentas AWS SDKs ou que melhor atenda às suas necessidades:

- [AWS SDKs Guia de referência de ferramentas e ferramentas](#)

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

Instalar, atualizar e desinstalar a AWS CLI

Este tópico fornece links para instalar, atualizar e desinstalar a versão original da AWS Command Line Interface (AWS CLI). A AWS CLI versão 1 atualmente é compatível, mas novos recursos adicionados à AWS CLI versão 2 não podem ser adicionados à AWS CLI versão 1. Para usar esses recursos, é necessário instalar a AWS CLI versão 2. Para obter mais informações sobre como instalar a versão 2, consulte [Instalar a AWS CLI versão 2](#).

Instruções para instalar, atualizar e desinstalar a AWS CLI:

- [Requisitos da versão do Python](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 no Amazon Linux](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 no Linux](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 no macOS](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 no Windows](#)
- [Instale e atualize a AWS CLI versão 1 em um ambiente virtual](#)

Requisitos da versão do Python

A AWS CLI versão 1 é criada com o SDK for Python e, portanto, requer a instalação de uma versão compatível do Python.

Matriz de suporte a versões do Python

Versão do AWS CLI	Versão do Python com suporte
1.32.0: atual	Python 3.8 e posterior
1.27.0–1.31.x	Python 3.7 e posterior
1.20.0–1.26.x	Python 3.6
1.19.0: 1.19.x	Python 2.7+, Python 3.6+
1.17 – 1.18.x	Python 2.7+, Python 3.4+
1.0 – 1.16.x	Python 2.6 e anteriores, Python 3.3 e anteriores

Para obter informações sobre as versões mais recentes da AWS CLI, consulte o [Log de alterações da AWS CLI versão 2](#) no GitHub.

Instale, atualize e desinstale a AWS CLI versão 1 no Amazon Linux

A AWS CLI versão 1 vem pré-instalada no Amazon Linux e no Amazon Linux 2. Verifique a versão atualmente instalada usando o comando a seguir.

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

Dependendo de quando você criou sua instância Amazon Linux, a AWS CLI versão 1 é pré-instalada usando um dos seguintes gerenciadores de pacotes:

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

Pré-requisitos

Você deve ter o Python 3.8 ou posterior instalado. Para obter instruções de instalação, consulte a página [Downloading Python](#) no Beginner's Guide do Python.

Matriz de suporte a versões do Python

AWS CLI versão	Versão do Python com suporte
1.32.0: atual	Python 3.8 e posterior
1.27.0–1.31.x	Python 3.7 e posterior
1.20.0–1.26.x	Python 3.6
1.19.0: 1.19.x	Python 2.7+, Python 3.6+
1.17 – 1.18.x	Python 2.7+, Python 3.4+
1.0 – 1.16.x	Python 2.6 e anteriores, Python 3.3 e anteriores

Instalar, atualizar e desinstalar usando pip

A maioria das instâncias do Amazon Linux usa pip para pré-instalar a AWS CLI versão 1.

Instale ou atualize a AWS CLI versão 1 no Amazon Linux usando pip

Para instalar a versão mais recente da AWS CLI versão 1 para o usuário atual, use as instruções a seguir.

1. Se você tiver o Python versão 3 ou posterior instalado, recomendamos usar o `pip3`. Use `pip3 install` para instalar ou atualizar para a versão mais recente da AWS CLI versão 1. Se você executar o comando em um [ambiente virtual Python \(venv\)](#), não será necessário usar a opção `--user`.

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

2. Verifique se a pasta que contém `aws` faz parte da variável `PATH`.
 - a. Encontre o script de perfil do shell no diretório de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

- Bash: `.bash_profile`, `.profile` ou `.bash_login`
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` ou `.login`.

- b. Adicione um comando de exportação ao final do script de perfil que é semelhante ao exemplo a seguir.

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

Esse comando insere o caminho `$HOME/.local/bin` neste exemplo na frente da variável `$PATH` existente.

- c. Recarregue o perfil em sua sessão atual para colocar essas alterações em vigor.

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

3. Para verificar se você está executando a nova versão, use o 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
```

Desinstale a AWS CLI versão 1 usando pip

Se você precisar desinstalar o AWS CLI, use `pip uninstall`.

```
$ pip3 uninstall awscli
```

Instalar, atualizar ou desinstalar usando yum

A maioria das instâncias do Amazon Linux 2 usa o yum para pré-instalar a AWS CLI versão 1.

Instale ou atualize a AWS CLI versão 1 no Amazon Linux usando o yum

Para instalar a versão mais recente da AWS CLI versão 1 disponível no Amazon Linux, execute o comando a seguir.

```
$ sudo yum install awscli
```

Para atualizar para a versão mais recente da AWS CLI versão 1 disponível no Amazon Linux, execute o comando a seguir.

```
$ sudo yum update awscli
```

Para verificar se você está executando a versão mais recente, use o 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
```

Desinstale a AWS CLI versão 1 usando o yum

Para desinstalar o AWS CLI, use `yum remove`.

```
$ sudo yum remove awscli
```

Solução de problemas de erros de AWS CLI instalação e desinstalação

Se você encontrar problemas após instalar ou desinstalar o AWS CLI, consulte as etapas de solução [Solucionar erros](#) de problemas. Para obter os passos mais relevantes para a solução de problemas, consulte [the section called “Erros de comando não encontrado”](#), [the section called “O comando “aws --version” retorna uma versão diferente da que você instalou”](#) e [the section called “O comando “aws --version” retorna uma versão após a desinstalação do AWS CLI”](#).

Instale, atualize e desinstale a AWS CLI versão 1 no Linux

Você pode instalar a AWS Command Line Interface (AWS CLI) versão 1 e suas dependências na maioria das distribuições Linux usando o gerenciador de pip pacotes ou o instalador incluído.

Embora o `awscli` pacote esteja disponível em repositórios para outros gerenciadores de pacotes, como `apt` e `yum`, eles não são produzidos, gerenciados ou suportados pelo AWS. Recomendamos que você instale o somente AWS CLI dos pontos de AWS distribuição oficiais, conforme documentado neste guia.

Seções

- [Pré-requisitos](#)
- [Instale e desinstale a AWS CLI versão 1 no Linux usando o instalador incluído](#)
- [Instale e desinstale a AWS CLI versão 1 usando pip](#)
- [Instale e desinstale a AWS CLI versão 1 usando o Snapcraft](#)
- [Adicione o executável da AWS CLI versão 1 ao seu caminho de linha de comando](#)
- [Solução de problemas de erros de AWS CLI instalação e desinstalação](#)

Pré-requisitos

Você deve ter o Python 3.8 ou posterior instalado. Para obter instruções de instalação, consulte a página [Downloading Python](#) no Beginner's Guide do Python.

Matriz de suporte a versões do Python

AWS CLI versão	Versão do Python com suporte
1.32.0: atual	Python 3.8 e posterior

AWS CLI versão	Versão do Python com suporte
1.27.0–1.31.x	Python 3.7 e posterior
1.20.0–1.26.x	Python 3.6
1.19.0: 1.19.x	Python 2.7+, Python 3.6+
1.17 – 1.18.x	Python 2.7+, Python 3.4+
1.0 – 1.16.x	Python 2.6 e anteriores, Python 3.3 e anteriores

Instale e desinstale a AWS CLI versão 1 no Linux usando o instalador incluído

No Linux ou macOS, é possível usar o instalador empacotado para instalar a versão 1 da AWS CLI. O pacote de instalador inclui todas as dependências do pacote e pode ser usado offline.

Note

O pacote de instalador fornecido não é compatível com a instalação em caminhos com espaços.

Tópicos

- [Instale a AWS CLI versão 1 usando o instalador incluído com sudo](#)
- [Instale a AWS CLI versão 1 usando o instalador incluído sem sudo](#)
- [Desinstale o instalador incluído na AWS CLI versão 1](#)

Instale a AWS CLI versão 1 usando o instalador incluído com **sudo**

As etapas a seguir permitem que você instale a AWS CLI versão 1 a partir da linha de comando em qualquer versão do Linux ou macOS.

Veja a seguir um resumo dos comandos de instalação explicados a seguir que você pode recortar e colar para executar como um único conjunto de comandos.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

```
$ 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
```

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte 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
```

Siga estas etapas na linha de comando para instalar a AWS CLI versão 1 usando o instalador incluído.

Para instalar a AWS CLI versão 1 usando o instalador incluído

1. Baixe o instalador incluído na AWS CLI versão 1 usando um dos métodos a seguir.
 - Faça download usando o comando `curl`.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte comando:

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

- Baixe usando o link direto.

Para a versão mais recente da AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte url <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

- Extraia os arquivos do pacote. Se você não tiver `unzip` para extrair os arquivos, use a distribuição do Linux incluída no gerenciador de pacote para fazer a instalação.

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

- Execute o programa de instalação. O instalador instala o AWS CLI at `/usr/local/aws` e cria o link simbólico `aws` no `/usr/local/bin` diretório. Usar a opção `-b` para criar um symlink elimina a necessidade de especificar o diretório de instalação na variável `$PATH` do usuário. Isso deve permitir que todos os usuários liguem para o AWS CLI entrando `aws` em qualquer diretório.

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

Por padrão, o script de instalação é executado sob a versão padrão do sistema do Python. Se você instalou uma versão alternativa do Python e deseja usar essa versão para instalar o AWS CLI, execute o script de instalação com essa versão pelo caminho absoluto até o executável do Python, da seguinte maneira.

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

- Verifique se AWS CLI está instalado corretamente.

```
$ 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 houver um erro, consulte [Solucionar problemas de erros AWS CLI](#).

Instale a AWS CLI versão 1 usando o instalador incluído sem **sudo**

Se você não tiver `sudo` permissões ou quiser instalar o AWS CLI somente para o usuário atual, poderá usar uma versão modificada dos comandos anteriores. Os dois primeiros comandos são os mesmos.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** awscli-bundle-1.16.312.zip resultaria no seguinte 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
```

Para instalar a AWS CLI versão 1 para o usuário atual

1. Baixe o instalador incluído na AWS CLI versão 1 de uma das seguintes formas.
 - Faça download usando o comando `curl`.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** awscli-bundle-1.16.312.zip resultaria no seguinte comando:

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

- Baixe usando o link direto.

Para a versão mais recente da AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** awscli-

`bundle-1.16.312.zip` resultaria no seguinte url <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

- Extraia os arquivos do pacote usando `unzip`. Se você não tiver `unzip`, use a distribuição do Linux incluída no gerenciador de pacote para fazer a instalação.

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

- Execute o programa de instalação. O instalador instala o AWS CLI at `/usr/local/aws` e cria o link simbólico `aws` no `/usr/local/bin` diretório. O comando usa o parâmetro `-b` para especificar o diretório no qual o instalador coloca o arquivo `symlink aws`. É necessário ter permissões de gravação na pasta especificada.

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

Isso AWS CLI instala o no local padrão (`~/local/lib/aws`) e cria um link simbólico (link simbólico) em `~/bin/aws`. Verifique se o `~/bin` está na sua variável de ambiente `PATH` para que o `symlink` funcione.

```
$ 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
```

- Certifique-se de que a AWS CLI versão 1 faça parte da sua `PATH` variável no diretório.
 - Encontre o script de perfil do shell em sua pasta de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

- Bash: `.bash_profile`, `.profile` ou `.bash_login`
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` ou `.login`.

- Adicione um comando de exportação ao final do script de perfil que é semelhante ao exemplo a seguir.

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

Esse comando insere o caminho `~/local/bin` neste exemplo na frente da variável `PATH` existente.

- c. Recarregue o perfil em sua sessão atual para colocar essas alterações em vigor.

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

5. Verifique se AWS CLI está instalado corretamente.

```
$ 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 houver um erro, consulte [Solucionar problemas de erros AWS CLI](#).

Desinstale o instalador incluído na AWS CLI versão 1

1. Se você instalou o AWS CLI usando o instalador incluído, siga estas instruções. O pacote de instalador não retira nada do diretório de instalação, exceto o symlink opcional. Portanto, a desinstalação é feita da mesma maneira destes dois itens.

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

2. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na `.aws` pasta.

Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o AWS CLI. Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `~/.aws/`. Se seu usuário tiver permissão de gravação nesse diretório, você não precisará usar `sudo`.

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

Instale e desinstale a AWS CLI versão 1 usando pip

Tópicos

- [Instalar o pip](#)
- [Instale e atualize a AWS CLI versão 1 usando pip](#)
- [Desinstale o AWS CLI pip usando](#)

Instalar o pip

Se você ainda não tem o pip instalado, você pode instalá-lo usando o script que o Python Packaging Authority fornece. Execute o `pip --version` para ver se a sua versão do Linux já inclui o Python e o pip. Se o Python versão 3 ou posterior estiver instalado, recomendamos que você use o comando `pip3`.

1. Use o comando `curl` a seguir para baixar o script de instalação. O comando a seguir usa o parâmetro `-O` (letra “O” maiúscula) para especificar que o arquivo baixado deve ser armazenado no diretório atual usando o mesmo nome que ele tem no host remoto.

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

2. Execute o script com o comando `python` ou `python3` para baixar e instalar a versão mais recente do pip e de outros pacotes de suporte necessários. Quando você inclui a chave `--user`, o script instala o pip no caminho `~/.local/bin`.

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

3. Verifique se o diretório que contém o pip faz parte da variável PATH.
 - a. Encontre o script de perfil do shell em sua pasta de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

- Bash: `.bash_profile`, `.profile` ou `.bash_login`
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` ou `.login`.

- b. Adicione um comando de exportação ao final do script de perfil que é semelhante ao exemplo a seguir.

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

Esse comando insere o caminho ~/.local/bin neste exemplo na frente da variável PATH existente.

- c. Recarregue o perfil em sua sessão atual para colocar essas alterações em vigor.

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

4. Para verificar se pip ou pip3 está instalado corretamente, execute o comando a seguir.

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

Instale e atualize a AWS CLI versão 1 usando pip

1. Use o comando pip ou pip3 para instalar ou atualizar a AWS CLI. Se você usar o Python versão 3 ou posterior, recomendamos usar o comando pip3. O --user switch pip instala o AWS CLI to ~/.local/bin.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente dois sinais de igual = e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** seria **==1.16.312** resultando no seguinte comando:

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

Note

Use as regras de aspas apropriadas para o terminal. Para usar o caractere =, você pode precisar usar aspas simples ou duplas para realizar o escape corretamente. O escape de exemplo a seguir usa aspas simples:

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

2. Verifique se AWS CLI está instalado corretamente.

```
$ 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 houver um erro, consulte [Solucionar problemas de erros AWS CLI](#).

Desinstale o AWS CLI pip usando

1. Se você instalou a AWS CLI versão 1 usando pip, você também deve desinstalar usando pip.

```
$ pip uninstall awscli
```

Se você estiver usando o Python versão 2 ou 3, talvez seja necessário usar o comando pip2 ou pip3. Use o `aws --version` comando para determinar a versão do Python associada à sua AWS CLI versão instalada 1.

```
$ pip3 uninstall awscli
```

Talvez seja necessário reiniciar a janela do prompt de comando ou o computador para remover todos os arquivos.

2. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na `.aws` pasta.

Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o AWS CLI. Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `~/.aws/`. Se seu usuário tiver permissão de gravação nesse diretório, você não precisará usar `sudo`.

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

Instale e desinstale a AWS CLI versão 1 usando o Snapcraft

Tópicos

- [Instalar snap](#)
- [Instale e atualize a AWS CLI versão 1 usando o snap](#)
- [Desinstale o AWS CLI usando o snap](#)

Instalar snap

Se ainda não tiver `snap` instalado, você pode instalá-lo usando as instruções fornecidas pela Canonical Snapcraft. Execute `snap version` para ver se sua versão do Linux já inclui `snap`.

1. Instale o Snapcraft na sua plataforma. Para obter informações sobre como instalar o Snapcraft, consulte [Instalando o daemon na documentação](#) do Snap.
2. Reinicie o sistema para que suas `PATH` variáveis sejam atualizadas corretamente. Se você estiver tendo problemas de instalação, siga as etapas em [Corrigir problemas comuns](#) na documentação do Snap.
3. Para verificar se `snap` está instalado corretamente, execute o comando a seguir.

```
$ snap version
```

Instale e atualize a AWS CLI versão 1 usando o snap

1. Execute o `snap install` comando a seguir para a AWS CLI versão 1.

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

Dependendo das suas permissões, talvez seja necessário adicionar algo `sudo` ao seu comando.

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

2. Verifique se AWS CLI está instalado corretamente.

```
$ 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 houver um erro, consulte [Solucionar problemas de erros AWS CLI](#).

Desinstale o AWS CLI usando o snap

1. Se você instalou a AWS CLI versão 1 usando `snap`, você também deve desinstalar usando `snap`.

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

Talvez seja necessário reiniciar a janela do prompt de comando ou o computador para remover todos os arquivos.

2. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na `.aws` pasta.

Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o AWS CLI. Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `~/.aws/`. Se você tiver permissões de gravação nesse diretório, não precisará usar `sudo`.

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

Adicione o executável da AWS CLI versão 1 ao seu caminho de linha de comando

Depois de instalar com `pip` ou `snap`, talvez seja necessário adicionar o `aws` executável à variável de `PATH` ambiente do seu sistema operacional.

Você pode verificar qual pasta `pip` instalou o AWS CLI in executando o comando a seguir.

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

Você pode fazer referência a isso como `~/.local/bin/` porque `/home/username` corresponde a `~` no Linux.

Se você omitiu a chave `--user` e, portanto, não a instalou no modo usuário, o arquivo executável poderá estar na pasta `bin` de sua instalação do Python. Se você não souber onde o Python está instalado, execute este comando.

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

A saída pode ser o caminho para um symlink, e não para o arquivo executável real. Execute `ls -al` para saber para onde ele aponta.

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

O `pip` instala programas na mesma pasta que contém o aplicativo Python. Adicione esta pasta à variável `PATH`.

Como modificar a variável **PATH**

1. Encontre o script de perfil do shell no diretório de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

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

- Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc`, ou `.login`
2. Adicione um comando de exportação ao script de perfil.

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

Este comando adiciona um caminho, `~/local/bin` neste exemplo, para a variável `PATH` atual.

3. Carregue o perfil atualizado em sua sessão atual.

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

Solução de problemas de erros de AWS CLI instalação e desinstalação

Se você encontrar problemas após instalar ou desinstalar o AWS CLI, consulte as etapas de solução [Solucionar erros](#) de problemas. Para obter os passos mais relevantes para a solução de problemas, consulte [the section called “Erros de comando não encontrado”](#), [the section called “O comando “aws --version” retorna uma versão diferente da que você instalou”](#) e [the section called “O comando “aws --version” retorna uma versão após a desinstalação do AWS CLI”](#).

Instale, atualize e desinstale a AWS CLI versão 1 no macOS

Você pode instalar a AWS Command Line Interface (AWS CLI) versão 1 e suas dependências no macOS usando o instalador incluído ou `pip`

Seções

- [Pré-requisitos](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 no macOS usando o instalador incluído](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 usando pip](#)
- [Solução de problemas de erros de AWS CLI instalação e desinstalação](#)

Pré-requisitos

Antes de instalar a AWS CLI versão 1 no macOS, verifique se você tem o Python 3.8 ou posterior instalado. Para obter instruções de instalação, consulte a página [Downloading Python](#) no Beginner's Guide do Python.

Matriz de suporte a versões do Python

AWS CLI versão	Versão do Python com suporte
1.32.0: atual	Python 3.8 e posterior
1.27.0–1.31.x	Python 3.7 e posterior
1.20.0–1.26.x	Python 3.6
1.19.0: 1.19.x	Python 2.7+, Python 3.6+
1.17 – 1.18.x	Python 2.7+, Python 3.4+
1.0 – 1.16.x	Python 2.6 e anteriores, Python 3.3 e anteriores

Instale, atualize e desinstale a AWS CLI versão 1 no macOS usando o instalador incluído

No Linux ou macOS, é possível usar o instalador empacotado para instalar a versão 1 da AWS Command Line Interface (AWS CLI). O pacote de instalador inclui todas as dependências do pacote e pode ser usado offline.

O pacote de instalador fornecido não é compatível com a instalação em caminhos com espaços.

Tópicos

- [Instale a AWS CLI versão 1 usando o instalador incluído com sudo](#)
- [Instale a AWS CLI versão 1 usando o instalador incluído sem sudo](#)
- [Desinstale o instalador incluído na AWS CLI versão 1](#)

Instale a AWS CLI versão 1 usando o instalador incluído com **sudo**

As etapas a seguir permitem que você instale a AWS CLI versão 1 a partir da linha de comando em qualquer versão do macOS.

Veja a seguir um resumo dos comandos de instalação que você pode recortar e colar para executar como um único conjunto de comandos.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

```
$ 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
```

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte 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
```

Para instalar a AWS CLI versão 1 usando o instalador incluído

1. Baixe o instalador incluído na AWS CLI versão 1 de uma das seguintes formas:

- Faça download usando o comando `curl`.

Para a versão mais recente da AWS CLI, use o seguinte bloco de comandos:

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

Para uma versão específica da AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte comando:

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

- Baixe usando o link direto.

Para a versão mais recente da AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte url <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Extraia (descompacte) os arquivos do pacote. Se você não tiver `unzip`, use o gerenciador de pacotes embutido da sua macOS distribuição para instalá-lo.

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

3. Execute o programa de instalação. O instalador instala o AWS CLI at `/usr/local/aws` e cria o link simbólico `aws` na `/usr/local/bin` pasta. Usar a opção `-b` para a criação de um symlink elimina a necessidade de especificar a pasta de instalação na variável `$PATH` do usuário. Isso deve permitir que todos os usuários chamem o AWS CLI entrando em `aws` qualquer diretório.

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

Por padrão, o script de instalação é executado sob a versão padrão do sistema do Python. Se você instalou uma versão alternativa do Python e deseja usá-la para instalar o AWS CLI, execute o script de instalação com essa versão pelo caminho absoluto até o executável do Python, da seguinte maneira.

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

4. Verifique se AWS CLI está instalado corretamente.

```
$ 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 houver um erro, consulte [Solucionar problemas de erros AWS CLI](#).

Instale a AWS CLI versão 1 usando o instalador incluído sem **sudo**

Se você não tiver sudo permissões ou quiser instalar o AWS CLI somente para o usuário atual, poderá usar uma versão modificada dos comandos anteriores. Os dois primeiros comandos são os mesmos.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** awscli-bundle-1.16.312.zip resultaria no seguinte 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
```

Para instalar a AWS CLI versão 1 para o usuário atual

1. Baixe o instalador incluído na AWS CLI versão 1 usando um dos seguintes métodos:

- Faça download usando o comando `curl`.

Para a versão mais recente da AWS CLI, use o seguinte bloco de comandos:

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

Para uma versão específica da AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** awscli-bundle-1.16.312.zip resultaria no seguinte comando:

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

- Baixe usando o link direto.

Para a versão mais recente da AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Para uma versão específica do AWS CLI, acrescente um hífen e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** `awscli-bundle-1.16.312.zip` resultaria no seguinte url <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Extraia os arquivos do pacote. Se você não tiver `unzip`, use a distribuição do Linux incluída no gerenciador de pacote para fazer a instalação.

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

3. Execute o programa de instalação. O instalador instala o AWS CLI at `/usr/local/aws` e cria o link simbólico `aws` no `/usr/local/bin` diretório. O comando usa o parâmetro `-b` para especificar o diretório no qual o instalador coloca o arquivo symlink `aws`. É necessário ter permissões de gravação no diretório especificado.

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

Isso AWS CLI instala o no local padrão (`~/local/lib/aws`) e cria um link simbólico (link simbólico) em `~/bin/aws`. Verifique se o `~/bin` está na sua variável de ambiente `$PATH` para que o symlink funcione.

```
$ 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. Verifique se a pasta na AWS CLI qual a versão 1 está instalada faz parte da sua `$PATH` variável.
 - a. Encontre o script de perfil do shell em sua pasta de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

- Bash: `.bash_profile`, `.profile` ou `.bash_login`
- Zsh: `.zshrc`
- Tcsh: `.tcshrc`, `.cshrc` ou `.login`.

- b. Adicione um comando de exportação ao final do script de perfil que é semelhante ao exemplo a seguir.

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

Esse comando insere o caminho ~/.local/bin neste exemplo na frente da variável PATH existente.

- c. Recarregue o perfil em sua sessão atual para colocar essas alterações em vigor.

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

5. Verifique se AWS CLI está instalado corretamente.

```
$ 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 houver um erro, consulte [Solucionar problemas de erros AWS CLI](#).

Desinstale o instalador incluído na AWS CLI versão 1

1. O instalador empacotado coloca tudo dentro do diretório de instalação, exceto o symlink opcional, portanto, para desinstalar, basta excluir esses dois itens.

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

2. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na .aws pasta.

Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o. AWS CLI Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `~/.aws/`. Se seu usuário tiver permissão de gravação nesse diretório, você não precisará usar `sudo`.

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

Instale, atualize e desinstale a AWS CLI versão 1 usando pip

Também é possível usar o `pip` diretamente para instalar a AWS CLI.

Tópicos

- [Instalar o pip](#)
- [Instale e atualize o AWS CLI usando pip](#)
- [Adicione o executável da AWS CLI versão 1 ao caminho da linha de comando do macOS](#)
- [Desinstale o AWS CLI pip usando](#)

Instalar o pip

Se você ainda não tem o `pip` instalado, você pode instalá-lo usando o script que o Python Packaging Authority fornece. Execute o `pip --version` para ver se a sua versão do Linux já inclui o Python e o `pip`. Se o Python versão 3 ou posterior estiver instalado, recomendamos que você use o comando `pip3`.

1. Use o comando `curl` a seguir para baixar o script de instalação. O comando a seguir usa o parâmetro `-O` (letra “O” maiúscula) para especificar que o arquivo baixado deve ser armazenado na pasta atual usando o mesmo nome que ele tem no host remoto.

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

2. Execute o script com o comando `python` ou `python3` para baixar e instalar a versão mais recente do `pip` e de outros pacotes de suporte necessários. Quando você inclui a chave `--user`, o script instala o `pip` no caminho `~/.local/bin`.

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

Instale e atualize o AWS CLI usando pip

1. Use o comando `pip` ou `pip3` para instalar a AWS CLI. Se você usar o Python versão 3 ou posterior, recomendamos usar o comando `pip3`.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente dois sinais de igual `=` e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão `1.16.312` seria `==1.16.312` resultando no seguinte comando:

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

Note

Use as regras de aspas apropriadas para o terminal. Para usar o caractere `=`, você pode precisar usar aspas simples ou duplas para realizar o escape corretamente. O escape de exemplo a seguir usa aspas simples:

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

2. Verifique se o AWS CLI está instalado corretamente.

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

Se o programa não for encontrado, [adicione-o ao caminho da linha de comando](#).

Adicione o executável da AWS CLI versão 1 ao caminho da linha de comando do macOS

Após a instalação com o `pip`, talvez seja necessário adicionar o programa `aws` à variável de ambiente `PATH` do seu sistema operacional. A localização do programa depende de onde o Python está instalado.

Example AWS CLI local de instalação - macOS com Python 3.6 e (modo de usuário) **pip**

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

Substitua a versão do Python que você tem pela versão do exemplo anterior.

Se você não souber onde o Python está instalado, execute `which python`.

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

A saída pode ser o caminho para um symlink, e não o programa executável real. Execute `ls -al` para saber para onde ele aponta.

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

O `pip` instala programas na mesma pasta que contém o aplicativo Python. Adicione esta pasta à variável `PATH`.

Como modificar a variável **PATH**

1. Encontre o script de perfil do shell no diretório de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

- Bash – `.bash_profile`, `.profile`, ou `.bash_login`
 - Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc`, ou `.login`
2. Adicione um comando de exportação ao script de perfil.

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

Este comando adiciona um caminho, `~/local/bin` neste exemplo, para a variável `PATH` atual.

3. Carregue o perfil atualizado em sua sessão atual.

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

Desinstale o AWS CLI pip usando

1. Se você instalou a AWS CLI versão 1 usando pip, você também deve desinstalar usando pip.

```
$ pip uninstall awscli
```

Se você estiver usando o Python versão 2 ou 3, talvez seja necessário usar o comando pip2 ou pip3. Use o `aws --version` comando para determinar a versão do Python associada à sua AWS CLI versão instalada 1.

```
$ pip3 uninstall awscli
```

Talvez seja necessário reiniciar a janela do prompt de comando ou o computador para remover todos os arquivos.

2. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na `.aws` pasta.

Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o AWS CLI. Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `~/.aws/`. Se seu usuário tiver permissão de gravação nesse diretório, você não precisará usar `sudo`.

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

Solução de problemas de erros de AWS CLI instalação e desinstalação

Se você encontrar problemas após instalar ou desinstalar o AWS CLI, consulte as etapas de solução [Solucionar erros](#) de problemas. Para obter os passos mais relevantes para a solução de problemas, consulte [the section called “Erros de comando não encontrado”](#), [the section called “O comando “aws --version” retorna uma versão diferente da que você instalou”](#) e [the section called “O comando “aws --version” retorna uma versão após a desinstalação do AWS CLI”](#).

Instale, atualize e desinstale a AWS CLI versão 1 no Windows

Você pode instalar a versão 1 do AWS Command Line Interface (AWS CLI) no Windows usando um instalador autônomo (recomendado) `ouipip`, que é um gerenciador de pacotes para Python.

Não inclua o símbolo de prompt (`C:\>`) ao digitar um comando. Eles são incluídos nas listas do programa para diferenciar comandos que você digita da saída retornada pela AWS CLI. O restante desse guia usa o símbolo de prompt genérico (`$`), exceto nos casos em que um comando é específico do Windows.

Tópicos

- [Instale, atualize e desinstale a AWS CLI versão 1 usando o MSI instalador](#)
- [Instale, atualize e desinstale a AWS CLI versão 1 usando Python e pip no Windows](#)
- [Adicione o executável da AWS CLI versão 1 ao seu caminho de linha de comando](#)
- [Solução de problemas de erros de AWS CLI instalação e desinstalação](#)

Instale, atualize e desinstale a AWS CLI versão 1 usando o MSI instalador

A AWS CLI versão 1 é compatível com o Windows XP ou posterior. Para usuários do Windows, o pacote de MSI instalação oferece uma maneira familiar e conveniente de instalar a AWS CLI versão 1 sem instalar nenhum outro pré-requisito.

Instale e atualize a AWS CLI versão 1 usando o MSI instalador

Verifique a página de [lançamentos](#) GitHub para ver quando a versão mais recente foi lançada. Quando novas atualizações forem lançadas, você deverá repetir o processo de instalação para instalar a versão mais recente da AWS CLI versão 1.

1. Baixe o MSI instalador apropriado:

- AWS CLI MSI instalador para Windows (64 bits): <https://s3.amazonaws.com/aws-cli/AWSCLI64PY3.msi>
 - AWS CLI MSI instalador para Windows (32 bits): [.msi https://s3.amazonaws.com/aws-cli/AWSCLI32PY3](https://s3.amazonaws.com/aws-cli/AWSCLI32PY3)
 - AWS CLI arquivo de configuração combinado para Windows: <https://s3.amazonaws.com/aws-cli/AWSCLISetup.exe> (inclui os MSI instaladores de 32 e 64 bits e instala automaticamente a versão correta)
2. Execute o MSI instalador baixado ou o arquivo de configuração.
 3. Siga as instruções da tela. Por padrão, a AWS CLI versão 1 é instalada em C:\Program Files\Amazon\AWSCLI (versão de 64 bits) ou C:\Program Files (x86)\Amazon\AWSCLI (versão de 32 bits).
 4. Para confirmar a instalação, use o comando `aws --version` em um prompt de comando (abra o menu Start (Iniciar) e pesquise cmd para iniciar um prompt de comando).

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

Se o Windows não conseguir encontrar o programa, talvez seja necessário fechar e reabrir o prompt de comando para atualizar o caminho ou [adicionar manualmente o diretório de instalação](#) à variável de PATH ambiente.

Desinstale a AWS CLI versão 1


Para usar as instruções de desinstalação a seguir, você precisa ter instalado a AWS CLI versão 1 com o MSI instalador ou o arquivo de configuração.

1. Abra Programas e Recursos seguindo um destes procedimentos:
 - Abra o Painel de Controle e selecione Programas e Recursos.
 - Abra um prompt de comando e execute o comando a seguir.

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

2. Selecione a entrada denominada AWS Command Line Interface e escolha Uninstall (Desinstalar) para executar o desinstalador.
3. Confirme que você deseja desinstalar AWS CLI o.

4. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na `.aws` pasta.

 Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o AWS CLI. Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `%UserProfile%\.aws`.


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

Instale, atualize e desinstale a AWS CLI versão 1 usando Python e pip no Windows

O Python Software Foundation fornece instaladores para Windows que incluem pip.

Pré-requisitos

Você deve ter o Python 3.8 ou posterior instalado. Para obter instruções de instalação, consulte a página [Downloading Python](#) no Beginner's Guide do Python.

 Warning

O Python 2.7 foi descontinuado pela [Python Software Foundation](#) em 1º de janeiro de 2020. A partir da AWS CLI versão 1.20.0, é necessária uma versão mínima do Python 3.6. Para usar a AWS CLI versão 1 com uma versão mais antiga do Python, você precisa instalar uma versão anterior da AWS CLI versão 1. Para ver a matriz de suporte da AWS CLI versão 1 do Python, consulte [the section called “Requisitos da versão do Python”](#)

Instale e atualize a AWS CLI versão 1 usando pip

1. Para instalar a AWS CLI versão 1, use o pip3 comando (se você usa o Python versão 3 ou posterior) ou o pip comando.

Para a versão mais recente do AWS CLI, use o seguinte bloco de comando:

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

Para uma versão específica do AWS CLI, acrescente um símbolo menor que < e o número da versão ao nome do arquivo. Neste exemplo, o nome do arquivo para a versão **1.16.312** seria **<1.16.312** resultando no seguinte comando:

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

2. Verifique se a AWS CLI versão 1 está instalada corretamente. Se não houver resposta, consulte a seção [Adicione o executável da AWS CLI versão 1 ao seu caminho de linha de comando](#).

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

Desinstale a AWS CLI versão 1 usando pip

1. Se você instalou a AWS CLI versão 1 usando pip, você também deve desinstalar usando pip.

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

Se você estiver usando o Python versão 2 ou 3, talvez seja necessário usar o comando pip2 ou pip3. Use o aws --version comando para determinar a versão do Python associada à sua AWS CLI versão instalada 1.

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

Talvez seja necessário reiniciar a janela do prompt de comando ou o computador para remover todos os arquivos.

2. (Opcional) Remova as informações compartilhadas AWS SDK e de AWS CLI configurações na `.aws` pasta.

⚠ Warning

Essas configurações e configurações de credenciais são compartilhadas entre todos AWS SDKs e o AWS CLI. Se você remover essa pasta, ela não poderá ser acessada por nenhuma AWS SDKs que ainda esteja no seu sistema.

A localização padrão da `.aws` pasta difere entre as plataformas. Por padrão, a pasta está localizada em `%UserProfile%\.aws`.

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

Adicione o executável da AWS CLI versão 1 ao seu caminho de linha de comando

Depois de instalar a AWS CLI versão 1 com `pip`, adicione o `aws` programa à variável de `PATH` ambiente do seu sistema operacional. Com uma MSI instalação, isso deve acontecer automaticamente. Mas se o comando `aws` não for executado após a instalação, talvez seja necessário configurá-lo manualmente.

1. Use o comando `where` para encontrar o local do arquivo `aws`. Por padrão, o comando `where` mostra onde um programa especificado é encontrado no `PATH` do sistema.

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

Os caminhos que aparecem dependem da plataforma e do método utilizado para instalar a AWS CLI. Os nomes de pasta que incluem os números de versão podem variar. Esses exemplos refletem o uso do Python versão 3.7. Substitua a versão pelo número de versão que você está usando, conforme necessário. Os caminhos típicos incluem o seguinte:

- Python 3 e **pip3** – `C:\Program Files\Python37\Scripts\`
- Python 3 e opção `--user` do **pip3** em versões anteriores do Windows – `%USERPROFILE%\AppData\Local\Programs\Python\Python37\Scripts`
- Python 3 e **pip3**: opção do usuário no Windows 10: `%USERPROFILE%\AppData\Roaming\Python\Python37\Scripts`

- MSIinstalador (64 bits) — C:\Program Files\Amazon\AWSCLI\bin
- MSIinstalador (32 bits) — C:\Program Files (x86)\Amazon\AWSCLI\bin

Use as etapas a seguir levando em conta se um caminho de arquivo é retornado.

A file path is returned

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

É possível encontrar onde o programa aws está instalado executando o comando a seguir.

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

A file path is NOT returned

Se o comando `where` retornar o erro a seguir, ele não estará no PATH do sistema e você não poderá executá-lo inserindo seu nome.

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

Nesse caso, execute o comando `where` com o parâmetro `/R path` para instruí-lo a pesquisar todas as pastas e adicione o caminho manualmente. Use a linha de comando ou o Explorador de Arquivos para descobrir onde ele está instalado no computador.

```
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. Pressione a tecla Windows e digite **environment variables**.
3. Escolha Edit environment variables for your account (Editar variáveis de ambiente para sua conta).
4. Escolha e PATH, em seguida, escolha Editar.

5. Adicione o caminho encontrado no campo Variable value (Valor da variável). Por exemplo, **C:\Program Files\Amazon\AWSCLI\bin\aws.exe**.
6. Escolha OK duas vezes para aplicar as novas configurações.
7. Feche todos os prompts de comando em execução e abra novamente a janela do prompt de comando.

Solução de problemas de erros de AWS CLI instalação e desinstalação

Se você encontrar problemas após instalar ou desinstalar o AWS CLI, consulte as etapas de solução [Solucionar erros](#) de problemas. Para obter os passos mais relevantes para a solução de problemas, consulte [the section called “Erros de comando não encontrado”](#), [the section called “O comando “aws --version” retorna uma versão diferente da que você instalou”](#) e [the section called “O comando “aws --version” retorna uma versão após a desinstalação do AWS CLI”](#).

Instale e atualize a AWS CLI versão 1 em um ambiente virtual

Você pode evitar conflitos de versão de requisitos com outros pip pacotes instalando a versão 1 do AWS Command Line Interface (AWS CLI) em um ambiente virtual.

Tópicos

- [Pré-requisitos](#)
- [Instale e atualize a AWS CLI versão 1 em um ambiente virtual](#)
- [Solução de problemas de erros de AWS CLI instalação e desinstalação](#)

Pré-requisitos

- Python 3.8 ou posterior. Para obter instruções de instalação, consulte a página [Downloading Python](#) no Beginner's Guide do Python.

Matriz de suporte a versões do Python

AWS CLI versão	Versão do Python com suporte
1.32.0: atual	Python 3.8 e posterior
1.27.0–1.31.x	Python 3.7 e posterior

AWS CLI versão	Versão do Python com suporte
1.20.0–1.26.x	Python 3.6
1.19.0: 1.19.x	Python 2.7+, Python 3.6+
1.17 – 1.18.x	Python 2.7+, Python 3.4+
1.0 – 1.16.x	Python 2.6 e anteriores, Python 3.3 e anteriores

- pip ou pip3 está instalado.

Instale e atualize a AWS CLI versão 1 em um ambiente virtual

1. Instale virtualenv usando o pip.

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

2. Crie um ambiente virtual e nomeie-o.

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

Como alternativa, você pode usar a opção `-p` para especificar uma versão do Python que não seja a padrão.

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

3. Ative seu novo ambiente virtual.

Linux ou macOS

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

Windows

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

O prompt mudará para mostrar que o ambiente virtual está ativo.

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

4. Instale ou atualize a AWS CLI versão 1 em seu ambiente virtual.

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

5. Verifique se a AWS CLI versão 1 está instalada corretamente.

```
$ 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. Use o comando `deactivate` para sair do ambiente virtual. Sempre que você iniciar uma nova sessão, deverá ativar novamente o ambiente.

Solução de problemas de erros de AWS CLI instalação e desinstalação

Se você encontrar problemas após instalar ou desinstalar o AWS CLI, consulte as etapas de solução [Solucionar erros](#) de problemas. Para obter os passos mais relevantes para a solução de problemas, consulte [the section called “Erros de comando não encontrado”](#), [the section called “O comando “aws --version” retorna uma versão diferente da que você instalou”](#) e [the section called “O comando “aws --version” retorna uma versão após a desinstalação do AWS CLI”](#).

Configurar o AWS CLI

Esta seção explica como definir as configurações que o AWS Command Line Interface (AWS CLI) usa para interagir AWS. Incluindo o seguinte:

- As credenciais identificam quem está chamando a API. As credenciais de acesso são usadas para criptografar a solicitação aos AWS servidores para confirmar sua identidade e recuperar as políticas de permissões associadas. Essas permissões determinam as ações que você pode realizar. Para obter informações sobre como configurar suas credenciais, consulte [Autenticação e credenciais de acesso](#).
- Outros detalhes de configuração para informar AWS CLI como processar solicitações, como o formato de saída padrão e a AWS região padrão.

Note

AWS exige que todas as solicitações recebidas sejam assinadas criptograficamente. Ele AWS CLI faz isso por você. A “assinatura” inclui uma data/time stamp. Portanto, você deve garantir que a data e a hora do seu computador estejam definidas corretamente. Caso contrário, e a data/hora na assinatura estiver muito distante da data/hora reconhecida pelo AWS serviço, AWS rejeitará a solicitação.

Precedência de credenciais e configurações

As credenciais e as definições de configuração estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. Certos locais têm precedência sobre outros. As AWS CLI credenciais e as configurações têm precedência na seguinte ordem:

1. [Opções da linha de comando](#): substituem as configurações em qualquer outro local, como nos parâmetros `--region`, `--output` e `--profile`.
2. [Variáveis de ambiente](#): você pode armazenar valores nas variáveis de ambiente do sistema.
3. [Assumir perfil](#): assumo as permissões de um perfil do IAM por meio da configuração ou do comando `aws sts assume-role`.

4. [Assumir perfil com identidade da web](#): assuma as permissões de um perfil do IAM usando uma identidade da web por meio da configuração ou do comando `aws sts assume-role`.
5. [Arquivo de credenciais](#): os arquivos `credentials` e `config` são atualizados quando você executa o comando `aws configure`. O arquivo `credentials` está localizado em `~/.aws/credentials` no Linux ou MacOS ou em `C:\Users\USERNAME\.aws\credentials` no Windows.
6. [Processo personalizado](#): obtenha suas credenciais de uma fonte externa.
7. [Arquivo de configuração](#): os arquivos `credentials` e `config` são atualizados quando você executa o comando `aws configure`. O arquivo `config` está localizado em `~/.aws/config` no Linux ou MacOS ou em `C:\Users\USERNAME\.aws\config` no Windows.
8. [Credenciais de container](#): você pode associar uma função do IAM a cada uma das suas definições de tarefa do Amazon Elastic Container Service (Amazon ECS). As credenciais temporárias para essa função estão disponíveis para os contêineres dessa tarefa. Para mais informações, consulte [Funções do IAM para Tarefas](#) no Guia de Desenvolvedor Amazon Elastic Container Service.
9. [Credenciais de perfil de instância](#): você pode associar um perfil do IAM a cada uma das suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2). As credenciais temporárias para essa função estão disponíveis para o código em execução na instância. As credenciais são fornecidas por meio do serviço de metadados do Amazon EC2. Para obter mais informações, consulte [Funções do IAM para o Amazon EC2](#) no Guia do usuário do Amazon EC2 e [Como usar perfis de instância](#) no Guia do usuário do IAM.

Tópicos adicionais nesta seção

- [the section called “Configurações do arquivo de configuração e credenciais”](#)
- [the section called “Variáveis de ambiente”](#)
- [the section called “Opções de linha de comando”](#)
- [the section called “Conclusão de comando”](#)
- [the section called “Repetições”](#)
- [the section called “Use um HTTP proxy”](#)

Configurações do arquivo de configuração e credenciais

Você pode salvar as definições de configuração usadas com frequência e credenciais em arquivos que são mantidos pela AWS CLI.

Os arquivos são divididos em `profiles`. Por padrão, o AWS CLI usa as configurações encontradas no perfil chamado `default`. Para usar outras configurações, você pode criar e fazer referência a perfis adicionais.

Você pode substituir uma configuração definindo uma das variáveis de ambiente com suporte ou usando um parâmetro de linha de comando. Para obter mais informações sobre a precedência de configuração, consulte [Configurar o AWS CLI](#).

Note

Para obter informações sobre como configurar suas credenciais, consulte [Autenticação e credenciais de acesso](#).

Tópicos

- [Formato dos arquivos de credenciais e configuração](#)
- [Onde as definições de configuração ficam armazenadas?](#)
- [Usar perfis nomeados](#)
- [Definir e visualizar as configurações usando comandos](#)
- [Definir novos exemplos de comandos de configuração e credenciais](#)
- [Configurações de arquivo config compatíveis](#)

Formato dos arquivos de credenciais e configuração

Os arquivos `config` e `credentials` são organizados em seções. As seções incluem perfis e serviços. Uma seção é um conjunto nomeado de configurações e continua até que outra linha de definição de seção seja encontrada. Vários perfis e seções podem ser armazenados nos arquivos `config` e `credentials`.

Esses arquivos são arquivos de texto simples que usam o seguinte formato:

- Os nomes das seções estão entre colchetes `[]`, como `[default]`, `[profile user1]` e `[sso-session]`.
- Todas as entradas em uma seção assumem a forma geral de `setting_name=value`.
- As linhas podem ser comentadas iniciando-as com um caractere de hashtag (`#`).


```
output=text
```

Amazon EC2 instance metadata credentials

Este exemplo é para as credenciais obtidas dos metadados da EC2 instância de hospedagem da Amazon. Para obter mais informações, consulte [the section called “Use metadados de EC2 instâncias da Amazon para credenciais AWS CLI”](#).

Arquivo de credenciais

O arquivo `credentials` não é usado para esse método de autenticação.

Arquivo de configuração

```
[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

Para evitar riscos de segurança, não use IAM usuários para autenticação ao desenvolver software específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como [AWS IAM Identity Center](#).

Este exemplo é para as credenciais de longo prazo do AWS Identity and Access Management. Para obter mais informações, consulte [the section called “IAMusuários”](#).

Arquivo de credenciais

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
```

```
aws_secret_access_key=wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

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

Arquivo de configuração

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

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

Para obter mais informações e métodos adicionais de autorização e credencial, consulte [the section called “IAMusuários”](#).

Tipo de seção: **services**

A `services` seção é um grupo de configurações que configura endpoints personalizados para AWS service (Serviço da AWS) solicitações. Depois, um perfil é vinculado a uma seção `services`.

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

A seção `services` é separada em subseções por linhas `<SERVICE> =`, onde `<SERVICE>` é a chave de identificação do AWS service (Serviço da AWS). O AWS service (Serviço da AWS) identificador é baseado no API modelo, substituindo todos os espaços `serviceId` por sublinhados e colocando todas as letras em minúsculas. Para obter uma lista de todas as chaves de identificação de serviço a serem usadas na seção `services`, consulte [Use endpoints no AWS CLI](#). A chave de identificação de serviço é seguida por configurações aninhadas, cada uma em sua própria linha e recuada por dois espaços.

O exemplo a seguir configura o endpoint a ser usado para solicitações feitas ao Amazon DynamoDB serviço no `my-services` seção que é usada no `dev` perfil. Todas as linhas imediatamente seguintes que estejam recuadas são incluídas nessa subseção e se aplicam a esse serviço.

```
[profile dev]
```

```
services = my-services

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

Para obter mais informações sobre endpoints específicos de serviço, consulte [Use endpoints no AWS CLI](#).

Se seu perfil tiver credenciais baseadas em função configuradas por meio de um `source_profile` parâmetro para IAM assumir a funcionalidade de função, ele usará SDK somente configurações de serviço para o perfil especificado. Ele não usa perfis com funções vinculadas a ele. Por exemplo, usando o seguinte arquivo compartilhado `config`:

```
[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 você usar o perfil B e fizer uma chamada em seu código para a AmazonEC2, o endpoint será resolvido como `https://profile-b-ec2-endpoint.aws`. Se o seu código fizer uma solicitação para qualquer outro serviço, a resolução do endpoint não seguirá nenhuma lógica personalizada. O endpoint não é resolvido para o endpoint global definido no perfil A. Para que um endpoint global tenha efeito para o perfil B, você precisaria configurar `endpoint_url` diretamente no perfil B.

Onde as definições de configuração ficam armazenadas?

Ele AWS CLI armazena informações confidenciais de credenciais que você especifica `aws configure` em um arquivo local chamado `credentials`, em uma pasta nomeada `.aws` em seu diretório inicial. As opções menos confidenciais de configuração especificadas com `aws configure` são armazenadas em um arquivo local chamado `config`, também armazenado na pasta `.aws`, no seu diretório inicial.

Armazenar credenciais no arquivo de configuração

Você pode manter todas as configurações do seu perfil em um único arquivo, pois eles AWS CLI podem ler as credenciais do `config` arquivo. Se houver credenciais nos dois arquivos para um perfil que compartilhe o mesmo nome, as chaves no arquivo de credenciais terão precedência. Sugerimos manter as credenciais nos arquivos `credentials`. Esses arquivos também são usados pelos kits de desenvolvimento de software de vários idiomas (SDKs). Se você usar um dos SDKs além do AWS CLI, confirme se as credenciais devem ser armazenadas em seu próprio arquivo.

O local do diretório inicial varia de acordo com o sistema operacional, mas é acessado usando as variáveis de ambiente `%UserProfile%` no Windows e `$HOME` ou `~` (til) em sistemas baseados em Unix. Você pode especificar um local não padrão para os arquivos configurando as variáveis de ambiente `AWS_CONFIG_FILE` e `AWS_SHARED_CREDENTIALS_FILE` para outro caminho local. Para mais detalhes, consulte [Variáveis de ambiente para configurar o AWS CLI](#).

Quando você usa um perfil compartilhado que especifica uma função AWS Identity and Access Management (IAM), ele AWS CLI chama a AWS STS `AssumeRole` operação para recuperar credenciais temporárias. Essas credenciais são armazenadas (no `~/.aws/cli/cache`). AWS CLI Os comandos subsequentes usam as credenciais temporárias em cache até que elas expirem e, nesse ponto, atualizam AWS CLI automaticamente as credenciais.

Usar perfis nomeados

Se nenhum perfil for definido explicitamente, será usado o perfil `default`.

Para usar um perfil designado, adicione a opção `--profile` *profile-name* para o comando. O exemplo a seguir lista todas as suas EC2 instâncias da Amazon usando as credenciais e configurações definidas no `user1` perfil.

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

Para usar um perfil designado para vários comandos, evite especificar o perfil ao configurar o comando em cada variável de ambiente `AWS_PROFILE` como o perfil padrão. É possível substituir essa configuração usando o parâmetro `--profile`.

Linux or macOS

```
$ export AWS_PROFILE=user1
```

Windows

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

O uso de [set](#) para definir uma variável de ambiente altera o valor usado até o final da sessão de prompt de comando atual ou até que você defina a variável como um valor diferente.

Usar [setx](#) para definir uma variável de ambiente altera o valor em todos os shells de comando criados depois de executar o comando. Isso não afeta nenhum shell de comando que já esteja em execução no momento em que você executa o comando. Feche e reinicie o shell de comando para ver os efeitos da alteração.

Configurar a variável de ambiente altera o perfil padrão até o final do seu shell sessão ou até que você defina a variável a um valor diferente. Você pode tornar as variáveis de ambiente persistentes em sessões futuras colocando-as no script de inicialização do shell. Para obter mais informações, consulte [Variáveis de ambiente para configurar o AWS CLI](#).

Definir e visualizar as configurações usando comandos

Existem várias formas de visualizar e definir as configurações usando comandos.

[aws configure](#)

Execute esse comando para definir e visualizar rapidamente as credenciais do , a região e o formato de saída. O exemplo a seguir mostra valores de exemplo.

```
$ 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](#)

Você pode definir quaisquer credenciais ou configurações usando `aws configure set`. Especifique o perfil que você deseja visualizar ou modificar com a configuração `--profile`.

Por exemplo, o comando a seguir define a `region` no perfil chamado `integ`.

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

Para remover uma configuração, use uma string vazia como o valor ou exclua manualmente a configuração nos arquivos `config` e `credentials` em um editor de texto.

```
$ aws configure set cli_pager "" --profile integ
```

aws configure get

Você pode recuperar quaisquer credenciais ou configurações definidas usando `aws configure get`. Especifique o perfil que você deseja visualizar ou modificar com a configuração `--profile`.

Por exemplo, o comando a seguir recupera a configuração `region` no perfil chamado `integ`.

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

Se a saída estiver vazia, a configuração não estará explicitamente definida e usará o valor padrão.

aws configure list

Para listar os dados de configuração, use o comando `aws configure list`. Esse comando lista as informações de perfil, chave de acesso, chave secreta e configuração da região usadas para o perfil especificado. Para cada item de configuração, ele mostra o valor, onde o valor da configuração foi recuperado e o nome da variável de configuração.

Por exemplo, se você fornecer o Região da AWS em uma variável de ambiente, esse comando mostrará o nome da região que você configurou, que esse valor veio de uma variável de ambiente e o nome da variável de ambiente.

Para métodos de credenciais temporários, como funções e IAM Identity Center, esse comando exibe a chave de acesso temporariamente em cache e a chave de acesso secreta é exibida.

```
$ aws configure list
Name                Value                Type                Location
----                -
```


Amazon EC2 instance metadata credentials

Este exemplo é para as credenciais obtidas dos metadados da EC2 instância de hospedagem da Amazon. Não há assistente para esse processo, portanto, cada valor é definido usando o comando `aws configure set`. Para obter mais informações, consulte [the section called “Use metadados de EC2 instâncias da Amazon para credenciais 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

Para evitar riscos de segurança, não use IAM usuários para autenticação ao desenvolver software específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como [AWS IAM Identity Center](#).

Este exemplo é para as credenciais de longo prazo do AWS Identity and Access Management. Para obter mais informações, consulte [the section called “IAMusuários”](#).

```
$ 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
```

Configurações de arquivo `config` compatíveis

Tópicos

- [Configurações globais](#)
- [Configurações de comando personalizadas do S3](#)

As configurações a seguir têm suporte no arquivo `config`. Os valores listados no perfil especificado (ou padrão) serão usados, a menos que eles sejam ignorados pela presença de uma variável de

ambiente ou uma opção de linha de comando com o mesmo nome. Para obter informações sobre a ordem de precedência das configurações, consulte [Configurar o AWS CLI](#)

Configurações globais

api_versions

Alguns AWS serviços mantêm várias API versões para oferecer suporte à compatibilidade com versões anteriores. Por padrão, AWS CLI os comandos usam a API versão mais recente disponível. Você pode especificar uma API versão a ser usada em um perfil incluindo a `api_versions` configuração no `config` arquivo.

Essa é uma configuração “aninhada” seguida por uma ou mais linhas recuadas, cada uma identificando um AWS serviço e a API versão a ser usada. Consulte a documentação de cada serviço para entender quais API versões estão disponíveis.

O exemplo a seguir mostra como especificar uma API versão para dois AWS serviços. Essas API versões são usadas somente para comandos executados sob o perfil que contém essas configurações.

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

Essa configuração não tem uma variável de ambiente nem parâmetro de linha de comando equivalente.

aws_access_key_id

Especifica a chave de AWS acesso usada como parte das credenciais para autenticar a solicitação de comando. Embora ela possa ser armazenada no arquivo `config`, recomendamos armazená-la no arquivo `credentials`.

Pode ser substituída pela variável de ambiente `AWS_ACCESS_KEY_ID`. Você não pode especificar o ID de chave de acesso como uma opção de linha de comando.

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

aws_secret_access_key

Especifica a chave AWS secreta usada como parte das credenciais para autenticar a solicitação de comando. Embora ela possa ser armazenada no arquivo `config`, recomendamos armazená-la no arquivo `credentials`.

Pode ser substituída pela variável de ambiente `AWS_SECRET_ACCESS_KEY`. Você não pode especificar a chave de acesso secreta como uma opção de linha de comando.

```
aws_secret_access_key = wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

aws_session_token

Especifica um token de AWS sessão. Um token de sessão só será necessário se você especificar manualmente credenciais de segurança temporárias. Embora ela possa ser armazenada no arquivo `config`, recomendamos armazená-la no arquivo `credentials`.

Pode ser substituída pela variável de ambiente `AWS_SESSION_TOKEN`. Você não pode especificar o token de sessão como uma opção de linha de comando.

```
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT  
+FvwwqKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/  
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

ca_bundle

Especifica um pacote de certificados CA (um arquivo com a `.pem` extensão) que é usado para verificar SSL certificados.

Pode ser substituído pela variável de ambiente [AWS_CA_BUNDLE](#) ou pela opção de linha de comando [--ca-bundle](#).

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

cli_follow_urlparam

Especifica se as AWS CLI tentativas de seguir URL links nos parâmetros da linha de comando começam com `http://` ou `https://`. Quando ativado, o conteúdo recuperado é usado como o valor do parâmetro em vez do URL.

- `true`: o valor padrão. Se especificado, todos os parâmetros de string que comecem com `http://` ou `https://` são obtidos e todo o conteúdo baixado é usado como o valor do parâmetro para o comando.
- `false` — Se especificado, o AWS CLI não trata valores de cadeia de caracteres de parâmetros que começam com `http://` ou de `https://` forma diferente de outras cadeias de caracteres.

Essa entrada não tem uma variável de ambiente equivalente nem uma opção de linha de comando.

```
cli_follow_urlparam = false
```

cli_history

Desabilitado por padrão. Essa configuração ativa o histórico de comandos para a AWS CLI. Depois de ativar essa configuração, ele AWS CLI registra o histórico dos aws comandos.

```
cli_history = enabled
```

Você pode listar seu histórico usando o comando `aws history list` e usar os `command_ids` resultantes no comando `aws history show` para obter detalhes. Para obter mais informações, consulte [aws history](#) no guia de referência da AWS CLI .

cli_timestamp_format

Especifica o formato dos valores de timestamp incluídos no resultado. Você pode especificar qualquer um dos seguintes valores:

- `iso8601` — O valor padrão para a AWS CLI versão 2. [Se especificado, o AWS CLI reformata todos os carimbos de data/hora de acordo com 8601. ISO](#)

ISOOs carimbos de data/hora formatados 8601 se parecem com os exemplos a seguir. O primeiro exemplo mostra a hora no [Tempo Universal Coordenado \(UTC\)](#) incluindo um Z após o horário. A data e a hora são separadas por T.

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

Para especificar um fuso horário diferente, em vez de Z, especifique um + ou - e o número de horas em que o fuso horário desejado está adiantado ou atrasado UTC, como um valor de dois

dígitos. O exemplo a seguir mostra o mesmo horário do exemplo anterior, mas ajustado para o horário padrão do Pacífico, que está oito horas atrasadoUTC.

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

- `wire` — O valor padrão para a AWS CLI versão 1. Se especificado, AWS CLI exibe todos os valores de timestamp exatamente como recebidos na resposta da HTTP consulta.

Essa entrada não tem uma variável de ambiente equivalente nem uma opção de linha de comando.

```
cli_timestamp_format = iso8601
```

credential_process

Especifica um comando externo que AWS CLI é executado para gerar ou recuperar credenciais de autenticação a serem usadas para esse comando. O comando deve retornar as credenciais em um formato específico. Para obter mais informações sobre como usar essa configuração, consulte [Credenciais de origem com um processo externo](#).

Essa entrada não tem uma variável de ambiente equivalente nem uma opção de linha de comando.

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

credential_source

Usado em EC2 instâncias ou contêineres da Amazon para especificar onde AWS CLI eles podem encontrar credenciais para usar para assumir a função que você especificou com o `role_arn` parâmetro. Não é possível especificar `source_profile` e `credential_source` no mesmo perfil.

Esse parâmetro pode ter um dos três valores:

- `Ambiente` — Especifica que AWS CLI é para recuperar as credenciais de origem das variáveis de ambiente.
- `Ec2 InstanceMetadata` — Especifica que a AWS CLI IAM função anexada ao [perfil da EC2 instância](#) deve ser usada para obter as credenciais de origem.
- `EcsContainer`— Especifica que deve usar a IAM função anexada ao ECS contêiner como credenciais de origem. AWS CLI


```
credential_source = Ec2InstanceMetadata
```

duration_seconds

Especifica a duração máxima da sessão da função, em segundos. Esse valor pode variar de 900 segundos (15 minutos) até o valor configurado de duração máxima da sessão para a função (que pode ser até 43200). Esse é um parâmetro opcional e, por padrão, o valor é definido como 3600 segundos.

endpoint_url

Especifica o endpoint usado para todas as solicitações de serviço. Se essa configuração for usada na seção [services](#) do arquivo config, o endpoint será usado somente para o serviço especificado.

O exemplo a seguir usa o endpoint global `http://localhost:1234` e um endpoint específico de serviço de `http://localhost:4567` para o Amazon S3.

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

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

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como `AWS_ENDPOINT_URL_DYNAMODB`.
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).

5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

ignore_configure_endpoint_urls

Se ativado, o AWS CLI ignora todas as configurações personalizadas de endpoint especificadas no arquivo `config`. Os valores válidos são **true** e **false**.

```
ignore_configure_endpoint_urls = true
```

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como `AWS_ENDPOINT_URL_DYNAMODB`.
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

external_id

Especifica um identificador exclusivo que é usado por terceiros para assumir uma função em suas contas de clientes. Isso é mapeado para o parâmetro `ExternalId` na operação `AssumeRole`. Esse parâmetro será necessário somente se a política de confiança para a função especificar um valor para `ExternalId`. Para obter mais informações, consulte [Como usar uma ID externa ao conceder acesso aos seus AWS recursos a terceiros](#) no Guia do IAM usuário.

max_attempts

Especifica um valor máximo de tentativas que o manipulador de AWS CLI repetição usa, em que a chamada inicial conta para o `max_attempts` valor que você fornece.

Você pode substituir esse valor usando a variável de ambiente `AWS_MAX_ATTEMPTS`.

```
max_attempts = 3
```

mfa_serial

O número de identificação de um MFA dispositivo a ser usado ao assumir uma função. Isso é obrigatório somente se a política de confiança da função que está sendo assumida incluir uma condição que exija MFA autenticação. O valor pode ser um número de série para um dispositivo de hardware (como `GAHT12345678`) ou um Amazon Resource Name (ARN) para um MFA dispositivo virtual (como `arn:aws:iam::123456789012:mfa/user`).

output

Especifica o formato de saída padrão para comandos solicitados usando esse perfil. Você pode especificar qualquer um dos seguintes valores:

- **json**— A saída é formatada como uma [JSON](#) string.
- **text**: a saída é formatada como várias linhas de valores de string separados por tabulação. Isso pode ser útil para passar a saída para um processador de texto, como `grep`, `sed` ou `awk`.
- **table**: a saída é formatada como uma tabela usando os caracteres `+|-` para formar as bordas da célula. Geralmente, a informação é apresentada em um formato "amigável", que é muito mais fácil de ler do que outros, mas não tão útil programaticamente.

Pode ser substituído pela variável de ambiente `AWS_DEFAULT_OUTPUT` ou pela opção de linha de comando `--output`.

```
output = table
```

parameter_validation

Especifica se o AWS CLI cliente tenta validar os parâmetros antes de enviá-los ao ponto final do AWS serviço.

- `true`: o valor padrão. Se especificado, o AWS CLI executa a validação local dos parâmetros da linha de comando.
- `false` — Se especificado, o AWS CLI não valida os parâmetros da linha de comando antes de enviá-los ao endpoint do AWS serviço.

Essa entrada não tem uma variável de ambiente equivalente nem uma opção de linha de comando.

```
parameter_validation = false
```

region

Especifica o Região da AWS envio de solicitações para comandos solicitados usando esse perfil.

- É possível especificar qualquer um dos códigos de região disponíveis para o serviço escolhido, conforme listado em [Regiões e endpoints da AWS](#) na Referência geral da Amazon Web Services.
- `aws_global` permite que você especifique o endpoint global para serviços que oferecem suporte a um endpoint global, além de endpoints regionais, como AWS Security Token Service (AWS STS e Amazon Simple Storage Service (Amazon S3).

Você pode substituir esse valor usando a variável de ambiente `AWS_DEFAULT_REGION` ou a opção da linha de comando `--region`.

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

retry_mode

Especifica qual modo de repetição usa. AWS CLI Existem três modos de repetição disponíveis: herdado (o modo usado por padrão), padrão e adaptive. Para obter mais informações sobre novas tentativas, consulte [AWS CLI tenta novamente](#).

Você pode substituir esse valor usando a variável de ambiente `AWS_RETRY_MODE`.

```
retry_mode = standard
```

role_arn

Especifica o Amazon Resource Name (ARN) de uma IAM função que você deseja usar para executar os AWS CLI comandos. Também é necessário especificar um dos seguintes parâmetros para identificar as credenciais que têm permissão para assumir essa função:

- `source_profile`
- `credential_source`

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

A variável de ambiente [AWS_ROLE_ARN](#) sobrescreve essa configuração.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called “Assumir a função com a identidade da web”](#).

role_session_name

Especifica o nome a ser associado à sessão da função. Esse valor é fornecido ao RoleSessionName parâmetro quando ele AWS CLI chama a AssumeRole operação e se torna parte da função assumida de usuárioARN: `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Esse parâmetro é opcional. Se você não fornecer esse valor, um nome de sessão será gerado automaticamente. Esse nome aparece nos logs do AWS CloudTrail para entradas associadas a essa sessão.

```
role_session_name = maria_garcia_role
```

A variável de ambiente [AWS_ROLE_SESSION_NAME](#) sobrescreve essa configuração.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called “Assumir a função com a identidade da web”](#).

services

Especifica a configuração de serviço a ser usada em seu perfil.

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

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

Para obter mais informações sobre a seção `services`, consulte [the section called “services”](#).

A variável de ambiente [AWS_ROLE_SESSION_NAME](#) sobrescreve essa configuração.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called “Assumir a função com a identidade da web”](#).

sdk_ua_app_id

Um único Conta da AWS pode ser usado por vários aplicativos de clientes para fazer chamadas para Serviços da AWS. O ID do aplicativo identifica qual aplicativo de origem fez um conjunto de chamadas usando um AWS service (Serviço da AWS). AWS SDKs e os serviços não usam nem interpretam esse valor a não ser para trazê-lo de volta às comunicações com o cliente. Por exemplo, esse valor pode ser incluído em e-mails operacionais para identificar com exclusividade quais dos seus aplicativos estão associados à notificação.

O ID do aplicativo é uma string com comprimento máximo de 50 caracteres. Letras, números e os seguintes caracteres especiais são permitidos: ! \$ % & * + - . , ^ _ ` | ~ Por padrão, nenhum valor é atribuído.

```
sdk_ua_app_id = prod1
```

Essa configuração pode ser substituída usando a variável de ambiente [AWS_SDK_UA_APP_ID](#). Você não pode definir esse valor como um parâmetro de linha de comando.

source_profile

Especifica um perfil nomeado com credenciais de longo prazo que a AWS CLI pode usar para assumir uma função que você especificou com o parâmetro `role_arn`. Não é possível especificar `source_profile` e `credential_source` no mesmo perfil.

```
source_profile = production-profile
```

sts_regional_endpoints

Especifica como AWS CLI determina o endpoint do AWS serviço que o AWS CLI cliente usa para se comunicar com o AWS Security Token Service (AWS STS). O valor padrão para a AWS CLI versão 1 é `legacy`.

É possível especificar um dos dois valores:

- **legacy**— Usa o STS endpoint global `sts.amazonaws.com`, para as seguintes AWS regiões: `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`, e `us-west-2`. Todas as demais Regiões usam automaticamente o respectivo endpoint regional.
- **regional**— AWS CLI Sempre usa o AWS STS endpoint da região atualmente configurada. Por exemplo, se o cliente estiver configurado para usar `us-west-2`, todas as chamadas AWS STS serão feitas para o endpoint regional `sts.us-west-2.amazonaws.com` em vez do `sts.amazonaws.com` endpoint global. Para enviar uma solicitação para o endpoint global enquanto a configuração é habilitada, você pode definir a Região como `aws-global`.

Essa configuração pode ser substituída usando a variável de ambiente

`AWS_STS_REGIONAL_ENDPOINTS`. Você não pode definir esse valor como um parâmetro de linha de comando.

use_dualstack_endpoint

Permite o uso de endpoints de pilha dupla para enviar solicitações. Para saber mais sobre endpoints de pilha dupla, que oferecem suporte tanto ao tráfego quanto ao IPv6 tráfego, consulte [Como IPv4 usar endpoints de pilha dupla do Amazon S3 no Guia do usuário do Amazon Simple Storage Service](#). Endpoints de pilha dupla estão disponíveis para alguns serviços em algumas regiões. Se não existir um endpoint de pilha dupla para o serviço ou Região da AWS se a solicitação falhar. Ela fica desabilitada por padrão.

Isso é mutuamente exclusivo com a configuração `use_accelerate_endpoint`.

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando `--endpoint-url`.

2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como `AWS_ENDPOINT_URL_DYNAMODB`.
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

use_fips_endpoint

Alguns AWS serviços oferecem endpoints compatíveis com o [Federal Information Processing Standard \(FIPS\) 140-2](#) em alguns. Regiões da AWS Quando o AWS serviço oferece suporte FIPS, essa configuração especifica qual FIPS endpoint eles AWS CLI devem usar. Ao contrário dos AWS endpoints padrão, os FIPS endpoints usam uma biblioteca TLS de software compatível com 140-2. FIPS Esses endpoints podem ser necessários por empresas que interagem com o governo dos Estados Unidos.

Se essa configuração estiver ativada, mas não existir um FIPS endpoint para o serviço em seu Região da AWS, o AWS comando poderá falhar. Nesse caso, especifique manualmente o endpoint a ser usado no comando usando a opção [--endpoint-url](#) ou use [endpoints específicos do serviço](#).

Para obter mais informações sobre como especificar FIPS endpoints por Região da AWS, consulte [FIPSEndpoints](#) by Service.

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como [AWS_ENDPOINT_URL_DYNAMODB](#).
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

[web_identity_token_file](#)

Especifica o caminho para um arquivo que contém um token de acesso OAuth 2.0 ou token de ID do OpenID Connect fornecido por um provedor de identidade. A AWS CLI carrega o conteúdo desse arquivo e o transmite como o argumento `WebIdentityToken` para a operação `AssumeRoleWithWebIdentity`.

A variável de ambiente [AWS_WEB_IDENTITY_TOKEN_FILE](#) sobrescreve essa configuração.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called "Assumir a função com a identidade da web"](#).

tcp_keepalive

Especifica se o AWS CLI cliente usa pacotes TCP keep-alive.

Essa entrada não tem uma variável de ambiente equivalente nem uma opção de linha de comando.

```
tcp_keepalive = false
```

Configurações de comando personalizadas do S3

O Amazon S3 oferece suporte a várias configurações que definem como ele AWS CLI executa as operações do Amazon S3. Algumas se aplicam a todos os comandos do S3 nos namespaces `s3api` e `s3`. Outros são especificamente para os comandos “personalizados” do S3 que abstraem operações comuns e fazem mais do que one-to-one mapear uma API operação. Os comandos de transferência `aws s3 cp`, `sync`, `mv` e `rm` têm configurações adicionais que você pode usar para controlar as transferências do S3.

Todas essas opções podem ser configuradas especificando a configuração `s3` aninhada em seu arquivo `config`. Depois, cada configuração é recuada em sua própria linha.

Note

Essas configurações são totalmente opcionais. Você deve ser capaz de usar com êxito os comandos de transferência `aws s3` sem definir nenhuma dessas configurações. Essas configurações são fornecidas para permitir que você ajuste a performance ou considere o ambiente específico no qual está executando esses comandos `aws s3`.

Essas configurações são todas definidas em uma chave `s3` de nível superior no arquivo `config`, conforme mostrado no exemplo a seguir para o perfil `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
```

As configurações a seguir se aplicam a qualquer comando do S3 nos namespaces `s3` ou `s3api`.

addressing_style

Especifica qual estilo de endereçamento usar. Isso controla se o nome do bucket está no nome do host ou faz parte do URL. Os valores válidos são: `path`, `virtual` e `auto`. O valor padrão é `auto`.

Há dois estilos de criação de um endpoint do Amazon S3. O primeiro é chamado `virtual` e inclui o nome do bucket como parte do nome do host. Por exemplo: `https://bucketname.s3.amazonaws.com`. Como alternativa, com o path estilo, você trata o nome do bucket como se fosse um caminho noURI; por exemplo, `https://s3.amazonaws.com/bucketname`. O valor padrão em CLI is to use `auto`, que tenta usar o `virtual` estilo sempre que possível, mas volta ao path estilo quando necessário. Por exemplo, se o nome do bucket não for DNS compatível, o nome do bucket não poderá fazer parte do nome do host e deverá estar no caminho. Com `auto`, CLI ele detectará essa condição e mudará automaticamente para o path estilo para você. Se você definir o estilo de endereçamento com `path`, deverá garantir que a AWS região que você configurou AWS CLI corresponda à região do seu bucket.

payload_signing_enabled

Especifica se as cargas sigv4 devem ser SHA256 assinadas. Por padrão, isso está desativado para uploads de streaming (`UploadPartPutObject`) durante o uso HTTPS. Por padrão, isso é definido como `false` para uploads de streaming (`UploadPartPutObject`), mas somente se a estiver presente (ele é gerado por padrão) e o endpoint usa `ContentMD5 HTTPS`.

Se definido como verdadeiro, as solicitações do S3 recebem validação adicional de conteúdo na forma de uma SHA256 soma de verificação que é calculada para você e incluída na assinatura da solicitação. Se for definido como `false`, a soma de verificação não será calculada. Desativar essa opção poderá ser útil para reduzir a sobrecarga de performance criada pela soma da soma de verificação.

use_accelerate_endpoint

Use o endpoint Amazon S3 Accelerate para todos os comandos `s3` e `s3api`. O valor padrão é falso. Isso é mutuamente exclusivo com a configuração `use_dualstack_endpoint`.

Se definido como verdadeiro, ele AWS CLI direciona todas as solicitações do Amazon S3 para S3 Accelerate o endpoint em `s3-accelerate.amazonaws.com`. Para usar esse endpoint, é necessário ativar o bucket para usar o S3 Accelerate. Todas as solicitações são enviadas usando o estilo virtual de endereçamento de bucket: `my-bucket.s3-accelerate.amazonaws.com`. As solicitações `ListBuckets`, `CreateBucket` e `DeleteBucket` não são enviadas ao endpoint do S3 Accelerate porque esse endpoint não oferece suporte a essas operações. Esse comportamento também poderá ser definido se o parâmetro `--endpoint-url` estiver definido como `https://s3-accelerate.amazonaws.com` ou `http://s3-accelerate.amazonaws.com` para qualquer comando `s3` ou `s3api`.

As seguintes configurações se aplicam somente aos comandos do conjunto de comandos do namespace `s3`.

max_bandwidth

Especifica a largura de banda máxima que pode ser consumida para carregar e baixar dados de e para o Amazon S3. O padrão é sem limite.

Isso limita a largura de banda máxima que os comandos do S3 podem usar para transferir dados de e para o Amazon S3. Esse valor se aplica apenas a uploads e downloads; ele não se aplica a cópias nem as exclui. O valor é expresso em bytes por segundo. O valor pode ser especificado como:

- Um valor inteiro. Por exemplo, `1048576` define o uso máximo da largura de banda como 1 megabyte por segundo.
- Um valor inteiro seguido por um sufixo de taxa. Você pode especificar sufixos de taxa usando: `KB/s`, `MB/s` ou `GB/s`. Por exemplo, `300KB/s`, `10MB/s`.

Em geral, recomendamos que você primeiro tente reduzir o consumo de largura de banda diminuindo `max_concurrent_requests`. Se isso não adequar o consumo de largura de banda limite para a taxa desejada, você poderá usar a configuração `max_bandwidth` para limitar ainda mais o consumo de largura de banda. Isso ocorre porque `max_concurrent_requests` controla o número de threads em execução no momento. Se, em vez disso, você baixar primeiro, `max_bandwidth` mas deixar uma configuração `max_concurrent_requests` alta, isso pode fazer com que threads tenham que esperar desnecessariamente. Isso pode levar ao excesso de consumo de recursos e tempos limite de conexão.

max_concurrent_requests

Especifica o número máximo de solicitações simultâneas. O valor padrão é 10.

Os comandos de transferência `aws s3` são multithread. Em um determinado momento, várias solicitações do Amazon S3 podem estar em execução. Por exemplo, quando você usa o comando `aws s3 cp localdir s3://bucket/ --recursive` para fazer upload de arquivos em um bucket do S3, eles AWS CLI podem fazer upload dos arquivos `localdir/file1`, `localdir/file2`, e `localdir/file3` em paralelo. A configuração `max_concurrent_requests` especifica o número máximo de operações de transferência que podem ser executadas ao mesmo tempo.

Talvez você precise alterar esse valor por alguns motivos:

- Diminuir esse valor: em alguns ambientes, o padrão de 10 solicitações simultâneas pode sobrecarregar um sistema. Isso pode causar erros de tempo limite de conexão ou diminuir a capacidade de resposta do sistema. Reduzir esse valor faz com que os comandos de transferência do S3 usem menos recursos. A desvantagem é que as transmissões do S3 podem levar mais tempo para serem concluídas. Reduzir esse valor poderá ser necessário se você usar uma ferramenta para limitar a largura de banda.
- Aumentar esse valor: em alguns cenários, talvez você queira que as transferências do Amazon S3 sejam concluídas com o máximo de rapidez possível, usando tanta largura de banda de rede quanto necessário. Nesse cenário, o número padrão de solicitações simultâneas pode não ser suficiente para utilizar toda a largura de banda de rede disponível. O aumento desse valor pode melhorar o tempo necessário para concluir uma transferência do Amazon S3.

max_queue_size

Especifica o número máximo de tarefas na fila de tarefas. O valor padrão é 1000.

O usa AWS CLI internamente um modelo em que enfileira tarefas do Amazon S3 que são então executadas por consumidores cujos números são limitados por `max_concurrent_requests`. Em geral, uma tarefa é mapeada para uma única operação do Amazon S3. Por exemplo, a tarefa pode ser `PutObjectTask`, `GetObjectTask` ou `UploadPartTask`. A taxa na qual as tarefas são adicionadas à fila pode ser muito mais rápida do que a taxa na qual os consumidores concluem as tarefas. Para evitar o crescimento não vinculado, o tamanho da fila de tarefas é limitada a um tamanho específico. Essa configuração altera o valor desse número máximo.

Em geral, você não precisa alterar essa configuração. Essa configuração também corresponde ao número de tarefas que ele conhece e que precisam ser executadas. AWS CLI Isso significa que, por padrão, eles só AWS CLI podem ver 1000 tarefas à frente. Aumentar esse valor significa que eles AWS CLI podem saber mais rapidamente o número total de tarefas necessárias, supondo que a taxa de filas seja mais rápida do que a taxa de conclusão da tarefa. A desvantagem é que um `max_queue_size` maior requer mais memória.

multipart_chunksize

Especifica o tamanho do fragmento que ele AWS CLI usa para transferências de várias partes de arquivos individuais. O valor padrão é de 8 MB, com um mínimo de 5 MB.

Quando uma transferência de arquivos excede o `multipart_threshold`, a AWS CLI divide o arquivo em blocos desse tamanho. Esse valor pode ser especificado usando a mesma sintaxe que `multipart_threshold`, como o número de bytes como um número inteiro ou usando um tamanho e um sufixo.

`multipart_threshold`

Especifica o limite de tamanho AWS CLI usado para transferências de várias partes de arquivos individuais. O valor padrão é de 8 MB.

Ao carregar, baixar ou copiar um arquivo, os comandos do Amazon S3 alternarão para operações multipart se o arquivo exceder esse tamanho. É possível especificar esse valor de duas formas:

- O tamanho do arquivo em bytes. Por exemplo, 1048576.
- O tamanho do arquivo com um sufixo de tamanho. Você pode usar KB, MB, GB ou TB. Por exemplo: 10MB, 1GB.

Note

O S3 pode impor restrições quanto a valores válidos que podem ser usados para operações multipart. Para obter mais informações, consulte a [documentação do Carregamento fracionado do S3](#) no Manual do usuário do Amazon Simple Storage Service.

Variáveis de ambiente para configurar o AWS CLI

As variáveis de ambiente fornecem outra forma de especificar opções e credenciais de configuração e podem ser úteis para criação de scripts.

Precedência de opções

- Se você especificar uma opção usando uma das variáveis de ambiente descritas nesse tópico, ela substituirá qualquer valor carregado de um perfil no arquivo de configuração.
- Se você especificar uma opção usando um parâmetro na linha de AWS CLI comando, ela substituirá qualquer valor da variável de ambiente correspondente ou de um perfil no arquivo de configuração.

Para obter mais informações sobre precedência e como AWS CLI determinar quais credenciais usar, consulte. [Configurar o AWS CLI](#)

Tópicos

- [Como definir variáveis de ambiente](#)

- [AWS CLI variáveis de ambiente suportadas](#)

Como definir variáveis de ambiente

Os exemplos a seguir mostram como configurar variáveis de ambiente para o usuário padrão.

Linux or macOS

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

Configurar a variável de ambiente altera o valor usado até o final da sua sessão de shell ou até que você defina a variável como um valor diferente. Você pode tornar as variáveis persistentes em sessões futuras definindo-as no script de inicialização do shell.

Windows Command Prompt

Como definir para todas as sessões

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

O uso de [setx](#) para definir uma variável de ambiente altera o valor usado na sessão de prompt de comando atual e todas as sessões de prompt de comando que você criar após a execução do comando. Não afeta outros shells de comando que já estejam em execução no momento em que você executar o comando. Talvez seja necessário reiniciar o terminal para que as configurações sejam carregadas.

Como definir somente para a sessão atual

O uso de [set](#) para definir uma variável de ambiente altera o valor usado até o final da sessão de prompt de comando atual ou até que você defina a variável como um valor diferente.

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

PowerShell

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

Se você definir uma variável de ambiente no PowerShell prompt, conforme mostrado nos exemplos anteriores, ela salvará o valor somente durante a sessão atual. Para tornar a configuração da variável de ambiente persistente em todas as sessões PowerShell e nas sessões do Prompt de Comando, armazene-a usando o aplicativo Sistema no Painel de Controle. Como alternativa, você pode definir a variável para todas as PowerShell sessões futuras adicionando-a ao seu PowerShell perfil. Consulte a [PowerShell documentação](#) para obter mais informações sobre como armazenar variáveis de ambiente ou persisti-las nas sessões.

AWS CLI variáveis de ambiente suportadas

O AWS CLI suporta as seguintes variáveis de ambiente.

AWS_ACCESS_KEY_ID

Especifica uma chave de AWS acesso associada a uma IAM conta.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `aws_access_key_id`. Você não pode especificar o ID de chave de acesso com uma opção de linha de comando.

AWS_CA_BUNDLE


Especifica o caminho para um pacote de certificados a ser usado para validação de HTTPS certificados.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil [ca_bundle](#). Você pode substituir essa variável de ambiente usando o parâmetro de linha de comando [--ca-bundle](#).

AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS

Se os buckets de origem e destino forem os mesmos ao usar o `s3 mv` comando personalizado, o arquivo ou objeto de origem poderá ser movido para si mesmo, o que pode resultar na exclusão acidental do arquivo ou objeto de origem. A variável de

`AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` ambiente e a `--validate-same-s3-paths` opção especificam se você deve validar seu ponto de acesso ARNs ou aliases de ponto de acesso em sua origem ou destino do Amazon S3. URIs

 Note

A validação do caminho para `s3 mv` exige API chamadas adicionais.

AWS_CONFIG_FILE

Especifica a localização do arquivo que ele AWS CLI usa para armazenar perfis de configuração. O caminho padrão é `~/.aws/config`.

Você não pode especificar esse valor em uma configuração de perfil nomeado nem usando um parâmetro de linha de comando.

AWS_DATA_PATH

Uma lista de diretórios adicionais a serem verificados fora do caminho de pesquisa integrado `~/.aws/models` ao carregar AWS CLI dados. A definição dessa variável de ambiente indica os diretórios adicionais a serem verificados primeiro antes de voltar para os caminhos de pesquisa internos. Várias entradas devem ser separadas com o caractere `.pathsep`, que é `:` no Linux ou no macOS e `;` no Windows.

AWS_DEFAULT_OUTPUT

Especifica o [formato de saída](#) a ser usado.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `output`. Você pode substituir essa variável de ambiente usando o parâmetro de linha de comando `--output`.

AWS_DEFAULT_REGION

`Default region name` identifica a AWS região para cujos servidores você deseja enviar suas solicitações por padrão. Normalmente, é a região mais próxima de você, mas pode ser qualquer região. Por exemplo, você pode digitar `us-west-2` para usar a região Oeste dos EUA (Oregon). Essa é a região para a qual todas as solicitações posteriores são enviadas, a menos que você especifique o contrário em um comando individual.

Note

Você deve especificar uma AWS região ao usar o AWS CLI, explicitamente ou definindo uma região padrão. Para obter uma lista das regiões disponíveis, consulte [Regiões e endpoints](#). Os designadores de região usados pelo AWS CLI são os mesmos nomes que você vê nos endpoints AWS Management Console URLs de serviço.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `region`. Você pode substituir essa variável de ambiente usando o parâmetro da linha de comando `--region`.

AWS_EC2_METADATA_DISABLED

Desativa o uso do serviço de metadados da EC2 instância Amazon (IMDS).

Se definido como verdadeiro, as credenciais ou a configuração do usuário (como a Região) não serão solicitadas IMDS.

AWS_ENDPOINT_URL

Especifica o endpoint usado para todas as solicitações de serviço.

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando `--endpoint-url`.
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como `AWS_ENDPOINT_URL_DYNAMODB`.
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.

7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

AWS_ENDPOINT_URL_<SERVICE>

Especifica um endpoint personalizado que é usado para um serviço específico, onde <SERVICE> é substituído pelo AWS service (Serviço da AWS) identificador. Por exemplo, Amazon DynamoDB tem um `serviceId` de [DynamoDB](#). Para esse serviço, a variável de URL ambiente do endpoint é `AWS_ENDPOINT_URL_DYNAMODB`.

Para obter uma lista de todas as variáveis de ambiente específicas do serviço, consulte [Lista de identificadores específicos de serviço](#).

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como `AWS_ENDPOINT_URL_DYNAMODB`.
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

AWS_IGNORE_CONFIGURED_ENDPOINT_URLS

Se ativado, o AWS CLI ignora todas as configurações personalizadas de endpoint. Os valores válidos são **true** e **false**.

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como [AWS_ENDPOINT_URL_DYNAMODB](#).
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

AWS_MAX_ATTEMPTS

Especifica um valor máximo de tentativas que o manipulador de AWS CLI repetição usa, em que a chamada inicial conta para o valor que você fornece. Para obter mais informações sobre novas tentativas, consulte [AWS CLI tenta novamente](#).

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `max_attempts`.

AWS_METADATA_SERVICE_NUM_ATTEMPTS

Ao tentar recuperar credenciais em uma EC2 instância da Amazon que tenha sido configurada com uma IAM função, ela AWS CLI tenta recuperar as credenciais uma vez do serviço de

metadados da instância antes de parar. Se você sabe que seus comandos serão executados em uma EC2 instância da Amazon, você pode aumentar esse valor para AWS CLI tentar novamente várias vezes antes de desistir.

AWS_METADATA_SERVICE_TIMEOUT

O número de segundos antes de uma conexão ao serviço de metadados da instância atingir o tempo limite. Ao tentar recuperar credenciais em uma EC2 instância da Amazon configurada com uma IAM função, a conexão com o serviço de metadados da instância expira após 1 segundo, por padrão. Se você sabe que está executando em uma EC2 instância da Amazon com uma IAM função configurada, você pode aumentar esse valor, se necessário.

AWS_PROFILE

Especifica o nome do AWS CLI perfil com as credenciais e as opções a serem usadas. Esse pode ser o nome de um perfil armazenado em um arquivo `credentials` ou `config` ou o valor `default` para usar o perfil padrão.

Se você especificar essa variável de ambiente, ela substituirá o comportamento de usar o perfil nomeado `[default]` no arquivo de configuração. Você pode substituir essa variável de ambiente usando o parâmetro de linha de comando `--profile`.

AWS_RETRY_MODE

Especifica qual modo de repetição usa. AWS CLI Existem três modos de repetição disponíveis: herdado (o modo usado por padrão), padrão e adaptivo. Para obter mais informações sobre novas tentativas, consulte [AWS CLI tenta novamente](#).

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `retry_mode`.

AWS_ROLE_ARN

Especifica o Amazon Resource Name (ARN) de uma IAM função com um provedor de identidade da web que você deseja usar para executar os AWS CLI comandos.

Usado com as variáveis de ambiente `AWS_WEB_IDENTITY_TOKEN_FILE` e `AWS_ROLE_SESSION_NAME`.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil [role_arn](#). Você não pode especificar um nome de sessão de função como um parâmetro de linha de comando.

Note

Essa variável de ambiente se aplica somente a uma função assumida com o provedor de identidade da Web, mas não se aplica à configuração do provedor de função assumida geral.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called “Assumir a função com a identidade da web”](#).

AWS_ROLE_SESSION_NAME

Especifica o nome a ser associado à sessão da função. Esse valor é fornecido ao RoleSessionName parâmetro quando ele AWS CLI chama a AssumeRole operação e se torna parte da função assumida de usuárioARN: `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Esse parâmetro é opcional. Se você não fornecer esse valor, um nome de sessão será gerado automaticamente. Esse nome aparece nos AWS CloudTrail registros das entradas associadas a essa sessão.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil [role_session_name](#).

Usado com as variáveis de ambiente AWS_ROLE_ARN e AWS_WEB_IDENTITY_TOKEN_FILE.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called “Assumir a função com a identidade da web”](#).

Note

Essa variável de ambiente se aplica somente a uma função assumida com o provedor de identidade da Web, mas não se aplica à configuração do provedor de função assumida geral.

AWS_SDK_UA_APP_ID

Um único Conta da AWS pode ser usado por vários aplicativos de clientes para fazer chamadas para Serviços da AWS. O ID do aplicativo identifica qual aplicativo de origem fez um conjunto de chamadas usando um AWS service (Serviço da AWS). AWS SDKse os serviços não usam

nem interpretam esse valor a não ser para trazê-lo de volta às comunicações com o cliente. Por exemplo, esse valor pode ser incluído em e-mails operacionais para identificar com exclusividade quais dos seus aplicativos estão associados à notificação.

Por padrão, não há valor.

O ID do aplicativo é uma string com comprimento máximo de 50 caracteres. Letras, números e os seguintes caracteres especiais são permitidos:

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

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil [sdk_ua_app_id](#). Você não pode especificar o ID do aplicativo como uma opção de linha de comando.

AWS_SECRET_ACCESS_KEY

Especifica a chave secreta associada à chave de acesso. Essencialmente, essa é a “senha” para a chave de acesso.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `aws_secret_access_key`. Você não pode especificar o ID chave de acesso secreta como uma opção de linha de comando.

AWS_SESSION_TOKEN

Especifica o valor de token de sessão que é necessário se você estiver usando credenciais de segurança temporárias recuperadas diretamente das operações do AWS STS . Para obter mais informações, consulte a [Seção de saída do comando `assume-role`](#) na Referência de comandos da AWS CLI .

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `aws_session_token`.

AWS_SHARED_CREDENTIALS_FILE

Especifica a localização do arquivo que o AWS CLI usa para armazenar as chaves de acesso. O caminho padrão é `~/.aws/credentials`.

Você não pode especificar esse valor em uma configuração de perfil nomeado nem usando um parâmetro de linha de comando.

AWS_STS_REGIONAL_ENDPOINTS

Especifica como AWS CLI determina o endpoint do AWS serviço que o AWS CLI cliente usa para se comunicar com o AWS Security Token Service (AWS STS). O valor padrão para a AWS CLI versão 1 é `legacy`.

É possível especificar um dos dois valores:

- **legacy**— Usa o STS endpoint `globalsts.amazonaws.com`, para as seguintes AWS regiões: `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`, e `us-west-2`. Todas as demais Regiões usam automaticamente o respectivo endpoint regional.
- **regional**— AWS CLI Sempre usa o AWS STS endpoint da região atualmente configurada. Por exemplo, se o cliente estiver configurado para usar `us-west-2`, todas as chamadas AWS STS serão feitas para o endpoint regional `sts.us-west-2.amazonaws.com` em vez do `sts.amazonaws.com` endpoint global. Para enviar uma solicitação para o endpoint global enquanto a configuração é habilitada, você pode definir a Região como `aws-global`.

AWS_USE_DUALSTACK_ENDPOINT

Permite o uso de endpoints de pilha dupla para enviar solicitações. Para saber mais sobre endpoints de pilha dupla, que oferecem suporte tanto ao tráfego quanto ao IPv6 tráfego, consulte [Como usar endpoints de pilha dupla do Amazon S3 no Guia do usuário do Amazon Simple Storage Service](#). Endpoints de pilha dupla estão disponíveis para alguns serviços em algumas regiões. Se não existir um endpoint de pilha dupla para o serviço ou Região da AWS se a solicitação falhar. Ela fica desabilitada por padrão.

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando `--endpoint-url`.
2. Se habilitada, a variável de ambiente global `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` ou a configuração do perfil `ignore_configure_endpoint_urls` para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço `AWS_ENDPOINT_URL_<SERVICE>`, como `AWS_ENDPOINT_URL_DYNAMODB`.

4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

AWS_USE_FIPS_ENDPOINT

Alguns AWS serviços oferecem endpoints compatíveis com o [Federal Information Processing Standard \(FIPS\) 140-2](#) em alguns. Regiões da AWS Quando o AWS serviço oferece suporte FIPS, essa configuração especifica qual FIPS endpoint eles AWS CLI devem usar. Ao contrário dos AWS endpoints padrão, os FIPS endpoints usam uma biblioteca TLS de software compatível com 140-2. FIPS Esses endpoints podem ser necessários por empresas que interagem com o governo dos Estados Unidos.

Se essa configuração estiver ativada, mas não existir um FIPS endpoint para o serviço em seu Região da AWS, o AWS comando poderá falhar. Nesse caso, especifique manualmente o endpoint a ser usado no comando usando a opção [--endpoint-url](#) ou use [endpoints específicos do serviço](#).

Para obter mais informações sobre como especificar FIPS endpoints por Região da AWS, consulte [FIPSEndpoints](#) by Service.

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como `AWS_ENDPOINT_URL_DYNAMODB`.

- Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
- O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
- O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
- Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
- Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

[AWS_WEB_IDENTITY_TOKEN_FILE](#)

Especifica o caminho para um arquivo que contém um token de acesso OAuth 2.0 ou token de ID do OpenID Connect fornecido por um provedor de identidade. A AWS CLI carrega o conteúdo desse arquivo e o transmite como o argumento `WebIdentityToken` para a operação `AssumeRoleWithWebIdentity`.

Usado com as variáveis de ambiente `AWS_ROLE_ARN` e `AWS_ROLE_SESSION_NAME`.

Se definida, essa variável de ambiente sobrescreverá o valor da configuração de perfil `web_identity_token_file`.

Para obter mais informações sobre o uso de identidades da Web, consulte [the section called “Assumir a função com a identidade da web”](#).

Note

Essa variável de ambiente se aplica somente a uma função assumida com o provedor de identidade da Web, mas não se aplica à configuração do provedor de função assumida geral.

Opções de linha de comando

No AWS CLI, as opções de linha de comando são parâmetros globais que você pode usar para substituir as configurações padrão, qualquer configuração de perfil correspondente ou configuração de variável de ambiente para esse único comando. Você não pode usar opções de linha de comando para especificar diretamente credenciais, embora seja possível especificar qual perfil usar.

Tópicos

- [Como usar as opções de linha de comando](#)
- [AWS CLI opções de linha de comando global suportadas](#)
- [Usos comuns das opções de linha de comando](#)

Como usar as opções de linha de comando

As opções de linha de comando são, na maioria, strings simples, como o nome do perfil `profile1` no exemplo a seguir:

```
$ aws s3 ls --profile profile1
example-bucket-1
example-bucket-2
...
```

Cada opção que obtém um argumento requer com um espaço ou sinal de igual (=) separando o argumento do nome da opção. Se o valor do argumento for uma string que contenha um espaço, coloque o argumento entre aspas. Para obter detalhes sobre tipos de argumento e formatação de parâmetros, consulte [Especifique os valores dos parâmetros para o AWS CLI](#).

AWS CLI opções de linha de comando global suportadas

No, AWS CLI você pode usar as seguintes opções de linha de comando para substituir as configurações padrão, qualquer configuração de perfil correspondente ou configuração de variável de ambiente para esse único comando.

`--ca-pacote` *<string>*

Especifica o pacote de certificados da autoridade de certificação (CA) a ser usado na verificação SSL de certificados.

Se definida, essa opção substituirá o valor da configuração de perfil `ca_bundle` e `AWS_CA_BUNDLE` da variável de ambiente.

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

Especifica o tempo de conexão de soquete máximo em segundos. Se o valor for definido como zero (0), a conexão de soquete aguardará indefinidamente (será bloqueada) e não atingirá o tempo limite.

--cli-read-timeout <integer>

Especifica o tempo de leitura de soquete máximo em segundos. Se o valor for definido como zero (0), a leitura do soquete aguardará indefinidamente (será bloqueada) e não atingirá o tempo limite.

--cor <string>

Especifica o suporte para saída de cores. Os valores válidos são on, off e auto. O valor padrão é auto.

--debug

Uma operação booliana que ativa o registro em log de depuração. AWS CLI Por padrão, fornece informações limpas sobre quaisquer sucessos ou falhas em relação aos resultados do comando na saída do comando. A opção `--debug` fornece os logs completos do Python. Isso inclui informações adicionais de diagnóstico de `stderr` sobre a operação do comando que podem ser úteis para a solução de problemas de um comando que gera resultados inesperados. Para visualizar facilmente os logs de depuração, sugerimos enviar os logs para um arquivo para que seja possível pesquisar as informações de forma mais fácil. Isso pode ser feito de uma das formas a seguir.

Para enviar somente as informações de diagnóstico de `stderr`, anexe `2> debug.txt`, onde `debug.txt` é o nome que você deseja usar para o seu arquivo de depuração:

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

Para enviar ambas a saída e as informações de diagnóstico de `stderr`, anexe `&> debug.txt`, onde `debug.txt` é o nome que você deseja usar para o seu arquivo de depuração:

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

--URL do ponto final <string>

Especifica o URL para o qual enviar a solicitação. Para a maioria dos comandos, o determina AWS CLI automaticamente URL com base no serviço selecionado e na AWS região especificada. No entanto, alguns comandos exigem que você especifique uma conta específica. URL Você também pode configurar alguns AWS serviços para [hospedar um endpoint diretamente em sua conta privada VPC](#), o que talvez precise ser especificado.

O exemplo de comando a seguir usa um endpoint personalizado do Amazon S3. URL

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

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando [--endpoint-url](#).
2. Se habilitada, a variável de ambiente global [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#) ou a configuração do perfil [ignore_configure_endpoint_urls](#) para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço [AWS_ENDPOINT_URL_<SERVICE>](#), como [AWS_ENDPOINT_URL_DYNAMODB](#).
4. Os valores fornecidos pelas variáveis de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#), [AWS_USE_FIPS_ENDPOINT](#) e [AWS_ENDPOINT_URL](#).
5. O valor do endpoint específico de serviço fornecido pela configuração [endpoint_url](#) em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração [endpoint_url](#) em um `profile` do arquivo compartilhado `config`.
7. Configurações [use_dualstack_endpoint](#), [use_fips_endpoint](#) e [endpoint_url](#).
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

`--no-paginate`

Um switch booleano que desativa as várias chamadas que ele AWS CLI faz automaticamente para receber todos os resultados do comando que cria a paginação da saída. Isso significa que apenas a primeira página da sua saída é exibida.

`--no-sign-request`

Um switch booleano que desativa a assinatura das HTTP solicitações no endpoint do AWS serviço. Isso evita que as credenciais sejam carregadas.

--no-verify-ssl

Por padrão, os AWS CLI usos SSL ao se comunicar com os AWS serviços. Para cada SSL conexão e chamada, ele AWS CLI verifica os SSL certificados. O uso dessa opção substitui o comportamento padrão de verificação SSL de certificados.

⚠ Warning

Essa opção não é uma prática recomendada. Se você usa `--no-verify-ssl`, seu tráfego entre seu cliente e AWS os serviços não está mais protegido. Isso significa que seu tráfego é um risco de segurança e vulnerável a man-in-the-middle explorações. Se estiver tendo problemas com certificados, é melhor resolvê-los. Para ver as etapas de resolução de problemas de certificados, consulte [the section called “SSLerros de certificado”](#).

--resultado <string>

Especifica o formato de saída a ser usado para este comando. Você pode especificar qualquer um dos seguintes valores:

- **json**— A saída é formatada como uma [JSON](#)string.
- **text**: a saída é formatada como várias linhas de valores de string separados por tabulação. Isso pode ser útil para passar a saída para um processador de texto, como `grep`, `sed` ou `awk`.
- **table**: a saída é formatada como uma tabela usando os caracteres `+|-` para formar as bordas da célula. Geralmente, a informação é apresentada em um formato "amigável", que é muito mais fácil de ler do que outros, mas não tão útil programaticamente.

--profile <string>

Especifica o [perfil nomeado](#) para usar esse comando. Para configurar perfis nomeados adicionais, você pode usar o comando `aws configure` com a opção `--profile`.

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

--consulta <string>

Especifica uma [JMESPathconsulta](#) a ser usada na filtragem dos dados de resposta. Para ter mais informações, consulte [AWS CLI Saída do filtro](#).

--região <string>

Especifica para qual AWS região enviar a AWS solicitação desse comando. Para obter uma lista de todas as regiões que você pode especificar, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

--version

Um switch booleano que exibe a versão atual do AWS CLI programa em execução.

Usos comuns das opções de linha de comando

Os usos comuns das opções de linha de comando incluem a verificação de seus recursos em várias regiões da AWS e a alteração do formato de saída para legibilidade ou a facilidade de uso ao criar scripts. Nos exemplos a seguir, executamos o comando `describe-instances` em cada região até descobrirmos em qual delas nossa instância está.

```
$ 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                          |||
...

```

Conclusão de comando

O AWS Command Line Interface (AWS CLI) inclui um recurso de conclusão de comando compatível com bash que permite usar a tecla Tab para concluir um comando parcialmente inserido. Na maioria dos sistemas, esse recurso deve ser configurado manualmente.

Tópicos

- [Como funciona](#)
- [Configuração do preenchimento automático de comando no Linux ou no macOS](#)
- [Configuração do preenchimento de comandos no Windows](#)

Como funciona

Ao inserir parcialmente um comando, um parâmetro ou uma opção, o recurso de conclusão de comando conclui automaticamente o comando ou exibe uma lista sugerida de comandos. Para solicitar a conclusão do comando, você insere parcialmente um comando e pressiona a tecla de conclusão, que normalmente é *Tab* na maioria das conchas.

Os exemplos a seguir mostram diferentes maneiras de usar a conclusão de comando:

- Insira parcialmente um comando e pressione *Tab* para exibir uma lista sugerida de comandos.

```
$ 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
```

- Insira parcialmente um parâmetro e pressione *Tab* para exibir uma lista sugerida de parâmetros.

```
$ 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
```

- Insira um parâmetro e pressione *Tab* para exibir uma lista sugerida de valores de recursos. Esse recurso está disponível somente na AWS CLI versão 2.

```
$ aws dynamodb db delete-table --table-name TAB
```


Table 1

Table 2

Table 3

Configuração do preenchimento automático de comando no Linux ou no macOS

Para configurar o preenchimento automático de comandos no Linux ou macOS, você deve conhecer o nome do shell que está usando e a localização do script `aws_completer`.

Note

A conclusão de comandos é automaticamente configurada e ativada por padrão nas EC2 instâncias da Amazon que executam o Amazon Linux.

Tópicos

- [Confirme se a pasta do completer está na variável path](#)
- [Habilitar a conclusão de comando](#)
- [Verifique o preenchimento de comandos](#)

Confirme se a pasta do completer está na variável path

Para que o AWS completador funcione com sucesso, ele `aws_completer` precisa estar no caminho do seu shell. O comando `which` pode verificar se o `completer` está na variável `path`.

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

Se o comando não conseguir encontrar o `completer`, use as etapas a seguir para adicionar a pasta do `completer` à variável `PATH`.

Etapas 1: localize o AWS completador

A localização do AWS completador pode variar dependendo do método de instalação usado.

- Package Manager - Programas como `pip`, `yumbrew`, e `apt-get` normalmente instalam o AWS completador (ou um link simbólico para ele) em um local de caminho padrão.

- Se você usou o pip sem o parâmetro `--user`, o caminho padrão é `/usr/local/bin/aws_completer`.
- Se você usou o pip com o parâmetro `--user`, o caminho padrão é `home/username/.local/bin/aws_completer`.
- Instalador empacotado: se você usou o instalador empacotado, o caminho padrão é `/usr/local/bin/aws_completer`.

Se tudo mais falhar, você pode usar o `find` comando para pesquisar o AWS completador em seu sistema de arquivos.

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

Etapa 2: Identificar o shell

Para identificar qual shell você está usando, é possível usar um dos seguintes comandos.

- `echo $ SHELL` — Exibe o nome do arquivo do programa do shell. Isso geralmente corresponde ao nome do shell em uso, a menos que você tenha iniciado um shell diferente após fazer o login.

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

- `ps`: exibe os processos em execução para o usuário atual. Um deles é o shell.

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

Etapa 3: Adicionar o completer ao caminho

1. Encontre o script de perfil do shell em sua pasta de usuário.

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

- Bash: `.bash_profile`, `.profile` ou `.bash_login`

- Zsh: `.zshrc`
 - Tcsh: `.tcshrc`, `.cshrc` ou `.login`
2. Adicione um comando de exportação ao final do script de perfil que é semelhante ao exemplo a seguir. Substitua `/usr/local/bin/` pela pasta que você descobriu na seção anterior.

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

3. Recarregue o perfil na sessão atual para colocar essas alterações em vigor. Substitua `.bash_profile` pelo nome do script de shell que você descobriu na primeira seção.

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

Habilitar a conclusão de comando

Depois de confirmar que o `completer` está no caminho, habilite a conclusão de comando executando o comando apropriado para o shell que você está usando. Você pode adicionar o comando ao perfil do shell para executá-lo cada vez que você abrir um novo shell. Em cada comando, substitua o `/usr/local/bin/` caminho com aquele encontrado em seu sistema em [Confirme se a pasta do `completer` está na variável `path`](#).

- **bash**: usa o comando interno `complete`.

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

Adicione o comando anterior ao `~/.bashrc` para executá-lo cada vez que você abrir um novo shell. O `~/.bash_profile` deve se basear em `~/.bashrc` para garantir que o comando também seja executado em shells de login.

- **zsh**: para executar o preenchimento de comandos, é necessário executar `bashcompinit` adicionando a seguinte linha de carregamento automático ao final do script do perfil `~/.zshrc`.

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

Para habilitar a conclusão de comando, use o comando interno `complete`.

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

Adicione os comandos anteriores ao `~/ .zshrc` para executá-los cada vez que você abrir um novo shell.

- **tcsh**: o preenchimento para `tcsh` utiliza um tipo de palavra e padrão para definir o comportamento do preenchimento.

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

Adicione o comando anterior ao `~/ .tcshrc` para executá-lo cada vez que você abrir um novo shell.

Depois de habilitar o preenchimento automático de comandos, [Verifique o preenchimento de comandos](#) está funcionando.

Verifique o preenchimento de comandos

Após habilitar o preenchimento de comandos, insira um comando parcial e pressione Tab para ver os comandos disponíveis.

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

Configuração do preenchimento de comandos no Windows

Note

Para obter informações sobre como PowerShell lidar com sua conclusão, incluindo suas várias chaves de conclusão, consulte [about_tab_Expansion no Microsoft Docs](#). PowerShell

Para habilitar a conclusão de comandos PowerShell no Windows, conclua as etapas a seguir em PowerShell.

1. Abra o arquivo `$PROFILE` com o comando a seguir.

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

Se você ainda não tiver um \$PROFILE, use o procedimento a seguir para criar um perfil de usuário.

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

Para obter mais informações sobre PowerShell perfis, consulte [Como usar perfis no Windows PowerShell ISE](#) no site do Microsoft Docs.

2. Para ativar o preenchimento de comandos, adicione o seguinte bloco de código ao seu perfil, salve e, em seguida, feche o arquivo.

```
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. Depois de habilitar a conclusão dos comandos, recarregue o shell, insira um comando parcial e pressione Tab para percorrer os comandos disponíveis.

```
$ aws sTab
```

```
$ aws s3
```

Para ver todos os comandos disponíveis para conclusão, insira um comando parcial e pressione Ctrl + Espaço.

```
$ aws sCtrl + Space  
s3          ses          sqs          sts          swf
```

s3api

sns

storagegateway support

AWS CLI tenta novamente

Este tópico descreve como as chamadas para AWS serviços AWS CLI podem falhar devido a problemas inesperados. Esses problemas podem ocorrer no lado do servidor ou podem falhar devido à limitação de taxa do serviço da AWS que você está tentando chamar. Esses tipos de falhas geralmente não exigem tratamento especial e a chamada é feita automaticamente novamente, em geral após um breve período de espera. AWS CLI Ele fornece muitos recursos para auxiliar na repetição de chamadas de clientes para AWS serviços quando esses tipos de erros ou exceções ocorrem.

Tópicos

- [Modos de novas tentativas disponíveis](#)
- [Configuração um modo de nova tentativa](#)
- [Visualização de logs de novas tentativas](#)

Modos de novas tentativas disponíveis

AWS CLI Tem vários modos para escolher, dependendo da sua versão:

- [Modo de novas tentativas herdado](#)
- [Modo de nova tentativa padrão](#)
- [Modo de nova tentativa adaptável](#)

Modo de novas tentativas herdado

O modo legado é o modo padrão usado pela AWS CLI versão 1. O modo herdado usa um manipulador de novas tentativas mais antigo e com funcionalidade limitada que inclui:

- Um valor padrão de 4 para o máximo de novas tentativas, totalizando 5 tentativas de chamada. Esse valor pode ser sobrescrito por meio do parâmetro de configuração `max_attempts`.
- No DynamoDB, o valor padrão máximo de novas tentativas é nove, totalizando dez tentativas de chamada. Esse valor pode ser sobrescrito por meio do parâmetro de configuração `max_attempts`.

- Novas tentativas para o seguinte número limitado de erros e exceções:
 - Erros gerais de soquete e conexão:
 - `ConnectionError`
 - `ConnectionClosedError`
 - `ReadTimeoutError`
 - `EndpointConnectionError`
 - Erros e exceções de controle de utilização ou limitação no lado do serviço:
 - `Throttling`
 - `ThrottlingException`
 - `ThrottledException`
 - `RequestThrottledException`
 - `ProvisionedThroughputExceededException`
- Tente novamente com vários códigos de HTTP status, incluindo 429, 500, 502, 503, 504 e 509.
- Qualquer tentativa de repetição incluirá um recuo exponencial por um fator de base 2.

Modo de nova tentativa padrão

O modo padrão é um conjunto padrão de regras de repetição em todo o mundo AWS SDKs com mais funcionalidade do que o legado. O modo padrão foi criado para a AWS CLI versão 2 e é transferido para a AWS CLI versão 1. A funcionalidade do modo padrão inclui:

- Um valor padrão de 2 para o máximo de novas tentativas, totalizando 3 tentativas de chamada. Esse valor pode ser sobrescrito por meio do parâmetro de configuração `max_attempts`.
- Novas tentativas para a seguinte lista estendida de erros/exceções:
 - Erros e exceções transientes
 - `RequestTimeout`
 - `RequestTimeoutException`
 - `PriorRequestNotComplete`
 - `ConnectionError`
 - `HTTPClientError`
 - Erros e exceções de controle de utilização ou limitação no lado do serviço:
 - `Throttling`

- `ThrottlingException`
 - `ThrottledException`
 - `RequestThrottledException`
 - `TooManyRequestsException`
 - `ProvisionedThroughputExceededException`
 - `TransactionInProgressException`
 - `RequestLimitExceeded`
 - `BandwidthLimitExceeded`
 - `LimitExceededException`
 - `RequestThrottled`
 - `SlowDown`
 - `EC2ThrottledException`
- Novas tentativas em códigos de erro não descritivos transientes. Especificamente, esses códigos de HTTP status: 500, 502, 503, 504.
 - Qualquer nova tentativa incluirá um recuo exponencial por um fator de base 2 para um tempo máximo de recuo de 20 segundos.

Modo de nova tentativa adaptável

Warning

O modo adaptativo é um experimental e está sujeito a modificações, tanto em suas características quanto em seu comportamento.

O modo de nova tentativa adaptativo é um modo de repetição experimental que inclui todos os recursos do modo padrão. Além dos recursos do modo padrão, o modo adaptativo também introduz limitação de taxa no lado do cliente através do uso de um bucket de token e variáveis de limite de taxa que são atualizadas dinamicamente com cada tentativa de repetição. Esse modo oferece flexibilidade nas novas tentativas do lado do cliente que se adaptam à resposta do estado de erro/exceção de um serviço. AWS

A cada nova tentativa de repetição, o modo adaptativo modifica as variáveis de limite de taxa com base no erro, na exceção ou no código de HTTP status apresentado na resposta do serviço. AWS Essas variáveis de limite de taxa são usadas para calcular uma nova taxa de chamada para o cliente. Cada HTTP resposta de exceção/erro ou falha (fornecida na lista acima) de um AWS serviço atualiza as variáveis de limite de taxa à medida que ocorrem novas tentativas até que o sucesso seja alcançado, o token bucket se esgote ou o valor máximo de tentativas configurado seja atingido.

Configuração um modo de nova tentativa

AWS CLI Isso inclui uma variedade de configurações de repetição e métodos de configuração a serem considerados ao criar seu objeto cliente.

Métodos de configuração disponíveis

No AWS CLI, os usuários podem configurar novas tentativas das seguintes maneiras:

- Variáveis de ambiente
- AWS CLI arquivo de configuração

Os usuários podem personalizar as seguintes opções de novas tentativas:

- Modo de repetição - especifica qual modo de repetição ele usa. AWS CLI Conforme descrito anteriormente existem três modos de repetição disponíveis: herdado (o modo usado por padrão), padrão e adaptativo. O valor padrão para a AWS CLI versão 1 é herdado, a .
- Máximo de tentativas - especifica o valor máximo de tentativas que o manipulador de AWS CLI repetição usa, em que a chamada inicial conta para o valor que você fornece. O valor padrão é 5.

Definição de uma configuração de novas tentativas em suas variáveis de ambiente

Para definir sua configuração de nova tentativa para o AWS CLI, atualize as variáveis de ambiente do seu sistema operacional.

As variáveis de ambiente de novas tentativas são:

- AWS_RETRY_MODE
- AWS_MAX_ATTEMPTS

Para obter mais informações sobre variáveis de ambiente, consulte [Variáveis de ambiente para configurar o AWS CLI](#).

Visualização de logs de novas tentativas

AWS CLI Ele usa a metodologia de repetição e o registro do Boto3. Você pode usar a opção `--debug` em qualquer comando para receber logs de depuração. Para obter informações sobre como usar a opção `--debug`, consulte [Opções de linha de comando](#).

Se você procurar “retry” em seus logs de depuração, encontrará as informações de repetição de que necessita. As entradas de log do cliente para tentativas de repetição dependem do modo de repetição ativado.

Modo herdado:

As mensagens de novas tentativas são geradas por `botocore.retryhandler`. Você verá uma de três mensagens:

- No retry needed
- Retry needed, action of: `<action_name>`
- Reached the maximum number of retry attempts: `<attempt_number>`

Modo padrão ou adaptativo:

As mensagens de novas tentativas são geradas por `botocore.retries.standard`. Você verá uma de três mensagens:

- No retrying request
- Retry needed, retrying request after delay of: `<delay_value>`
- Retry needed but retry quota reached, not retrying request

Para ver o arquivo de definição completo das novas tentativas de `botocore`, consulte [_retry.json](#) no repositório `botocore`. GitHub

Use um HTTP proxy

Para acessar AWS por meio de servidores proxy, você pode configurar as variáveis de ambiente `HTTPS_PROXY` `HTTP_PROXY` e com os nomes de DNS domínio ou endereços IP e números de porta que seus servidores proxy usam.

Tópicos

- [Como usar os exemplos da](#)
- [Autenticar para um proxy](#)
- [Usando um proxy em EC2 instâncias da Amazon](#)
- [Solução de problemas](#)

Como usar os exemplos da

Note

Os exemplos a seguir mostram o nome da variável de ambiente com todas as letras maiúsculas. No entanto, se você especificar uma variável duas vezes usando letras maiúsculas e minúsculas, as minúsculas terão precedência. Recomendamos que você defina cada variável somente uma vez para evitar confusão e comportamento inesperado do sistema.

Os exemplos a seguir mostram como você pode usar o endereço IP explícito do seu proxy ou um DNS nome que se resolva para o endereço IP do seu proxy. Também pode ser seguido por uma vírgula e o número da porta para a qual as consultas devem ser enviadas.

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

Como definir para todas as sessões

```
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
```

O uso de [setx](#) para definir uma variável de ambiente altera o valor usado na sessão de prompt de comando atual e todas as sessões de prompt de comando que você criar após a execução do comando. Não afeta outros shells de comando que já estejam em execução no momento em que você executar o comando.

Como definir somente para a sessão atual

O uso de [set](#) para definir uma variável de ambiente altera o valor usado até o final da sessão de prompt de comando atual ou até que você defina a variável como um valor diferente.

```
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
```

Autenticar para um proxy

Note

AWS CLI Não oferece suporte a NTLM proxies. [Se você usa um proxy de protocolo NTLM ou Kerberos, talvez consiga se conectar por meio de um proxy de autenticação como o Cntlm.](#)

O AWS CLI suporta autenticação HTTP básica. Especifique o nome de usuário e a senha no proxyURL, da seguinte forma.

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

Como definir para todas as sessões

```
C:\> setx HTTP_PROXY http://username:password@proxy.example.com:1234
C:\> setx HTTPS_PROXY http://username:password@proxy.example.com:5678
```

Como definir somente para a sessão atual

```
C:\> set HTTP_PROXY=http://username:password@proxy.example.com:1234
C:\> set HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

Usando um proxy em EC2 instâncias da Amazon

Se você configurar um proxy em uma EC2 instância da Amazon lançada com uma IAM função anexada, assegure-se de isentar o endereço usado para acessar os [metadados da instância](#). Para fazer isso, defina a variável de ambiente NO_PROXY como o endereço IP do serviço de metadados da instância 169.254.169.254. Esse endereço não varia.

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx NO_PROXY 169.254.169.254
```

Como definir somente para a sessão atual

```
C:\> set NO_PROXY=169.254.169.254
```

Solução de problemas

Se você encontrar problemas com o AWS CLI, consulte as etapas [Solucionar erros](#) de solução de problemas. Para obter as etapas mais relevantes de solução de problemas, consulte [the section called “SSLerros de certificado”](#).

Use endpoints no AWS CLI

Para se conectar programaticamente a um AWS service (Serviço da AWS), você usa um endpoint. Um endpoint é o ponto URL de entrada para um serviço AWS web. O AWS Command Line Interface (AWS CLI) usa automaticamente o endpoint padrão para cada serviço em um Região da AWS, mas você pode especificar um endpoint alternativo para suas API solicitações.

Tópicos de endpoint

- [Definir o endpoint para um único comando](#)
- [Defina um endpoint global para todos Serviços da AWS](#)
- [Definido para usar FIPs endpoints para todos Serviços da AWS](#)
- [Configurado para usar os endpoints de pilha dupla para todos os Serviços da AWS](#)
- [Definir endpoints específicos de serviço](#)
 - [Endpoints específicos de serviço: variáveis de ambiente](#)
 - [Endpoints específicos de serviço: arquivo compartilhado config](#)
 - [Endpoints específicos de serviço: lista de identificadores específicos de serviço](#)
- [Precedência de configurações e definições do endpoint](#)

Definir o endpoint para um único comando

Para substituir qualquer configuração de endpoint ou variável de ambiente referente a um único comando, use a opção de linha de comando `--endpoint-url`. O exemplo de comando a seguir usa um endpoint personalizado do Amazon S3. URL

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

Defina um endpoint global para todos Serviços da AWS

Para encaminhar solicitações de todos os serviços para um endpoint personalizadoURL, use uma das seguintes configurações:

- Variáveis de ambiente:
 - [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#)- Ignore o endpoint URLs configurado.

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

Como definir somente para a sessão atual

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS_ENDPOINT_URL](#) - Defina o endpoint URL global.

Linux or macOS

```
$ export AWS_ENDPOINT_URL=http://localhost:4567
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx AWS_ENDPOINT_URL http://localhost:4567
```

Como definir somente para a sessão atual

```
C:\> set AWS_ENDPOINT_URL=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL="http://localhost:4567"
```

• O arquivo config:

Defina um endpoint global para todos Serviços da AWS

- [ignore_configure_endpoint_urls](#)- Ignore o endpoint URLs configurado.

```
ignore_configure_endpoint_urls = true
```

- [endpoint_url](#)- Defina o endpoint URL global.

```
endpoint_url = http://localhost:4567
```

Os endpoints específicos de serviço e a opção de linha de comando `--endpoint-url` substituem qualquer endpoint global.

Definido para usar FIPs endpoints para todos Serviços da AWS

Para rotear solicitações para que todos os serviços FIPs usem endpoints, use uma das seguintes opções:

- variável de ambiente [AWS_USE_FIPS_ENDPOINT](#).

Linux or macOS

```
$ export AWS_USE_FIPS_ENDPOINT=true
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx AWS_USE_FIPS_ENDPOINT true
```

Como definir somente para a sessão atual

```
C:\> set AWS_USE_FIPS_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_FIPS_ENDPOINT="true"
```

- configuração do arquivo [use_fips_endpoint](#).

```
use_fips_endpoint = true
```


Alguns AWS serviços oferecem endpoints compatíveis com o [Federal Information Processing Standard \(FIPS\) 140-2](#) em alguns. Regiões da AWS Quando o AWS serviço oferece suporte FIPS, essa configuração especifica qual FIPS endpoint eles AWS CLI devem usar. Ao contrário dos AWS endpoints padrão, os FIPS endpoints usam uma biblioteca TLS de software compatível com 140-2. FIPS Esses endpoints podem ser necessários por empresas que interagem com o governo dos Estados Unidos.

Se essa configuração estiver ativada, mas não existir um FIPS endpoint para o serviço em sua Região da AWS, o AWS comando poderá falhar. Nesse caso, especifique manualmente o endpoint a ser usado no comando usando a opção `--endpoint-url` ou use [endpoints específicos do serviço](#).

Para obter mais informações sobre como especificar FIPS endpoints por Região da AWS, consulte [FIPSEndpoints](#) by Service.

Configurado para usar os endpoints de pilha dupla para todos os Serviços da AWS

Para rotear solicitações para todos os serviços usarem endpoints de pilha dupla, use uma das seguintes configurações:

- variável de ambiente [AWS_USE_DUALSTACK_ENDPOINT](#).

Linux or macOS

```
$ export AWS_USE_DUALSTACK_ENDPOINT=true
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
```

Como definir somente para a sessão atual

```
C:\> set AWS_USE_DUALSTACK_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_DUALSTACK_ENDPOINT="true"
```

- configuração do arquivo [use_dualstack_endpoint](#).

```
use_dualstack_endpoint = true
```

Permite o uso de endpoints de pilha dupla para enviar solicitações. AWS Para saber mais sobre endpoints de pilha dupla, que oferecem suporte tanto ao tráfego quanto ao IPv6 tráfego, consulte Como IPv4 usar endpoints de [pilha dupla do Amazon S3 no Guia do usuário do Amazon Simple Storage Service](#). Endpoints de pilha dupla estão disponíveis para alguns serviços em algumas regiões. Se não existir um endpoint de pilha dupla para o serviço ou Região da AWS se a solicitação falhar. Ela fica desabilitada por padrão.

Definir endpoints específicos de serviço

A configuração de endpoint específico do serviço oferece a opção de usar um endpoint persistente de sua escolha para solicitações. AWS CLI Essas configurações oferecem flexibilidade para oferecer suporte a endpoints locais, VPC endpoints e ambientes de AWS desenvolvimento local de terceiros. Diferentes endpoints podem ser usados para ambientes de teste e produção. Você pode especificar um endpoint URL para cada pessoa Serviços da AWS.

Os endpoints específicos de serviço podem ser designados das seguintes maneiras:

- A opção de linha de comando [--endpoint-url](#) para um único comando.
- Variáveis de ambiente:
 - [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#)- Ignore todos os endpoints configuradosURLs, a menos que especificado na linha de comando.
 - [AWS_ENDPOINT_URL_<SERVICE>](#): especifica um endpoint personalizado usado para um serviço específico, onde <SERVICE> é substituído pelo identificador AWS service (Serviço da AWS) . Para todas as variáveis específicas de serviço, consulte [the section called “Lista de identificadores específicos de serviço”](#).
- Arquivo config:
 - [ignore_configure_endpoint_urls](#)- Ignore todos os endpoints configuradosURLs, a menos que especificado usando variáveis de ambiente ou na linha de comando.
 - A seção [services](#) do arquivo config combinada com a configuração do arquivo [endpoint_url](#).

Tópicos de endpoints específicos de serviço:

- [Endpoints específicos de serviço: variáveis de ambiente](#)
- [Endpoints específicos de serviço: arquivo compartilhado config](#)
- [Endpoints específicos de serviço: lista de identificadores específicos de serviço](#)

Endpoints específicos de serviço: variáveis de ambiente

As variáveis de ambiente substituem as configurações no arquivo de configuração, mas não substituem as opções especificadas na linha de comando. Use variáveis de ambiente se quiser que todos os perfis usem os mesmos endpoints no dispositivo.

Veja a seguir as variáveis de ambiente específicas de serviço:

- [AWS_IGNORE_CONFIGURED_ENDPOINT_URLS](#)- Ignore todos os endpoints configurados URLs, a menos que especificado na linha de comando.

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

Como definir somente para a sessão atual

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS_ENDPOINT_URL_<SERVICE>](#)- Especifica um endpoint personalizado que é usado para um serviço específico, onde <SERVICE> é substituído pelo AWS service (Serviço da AWS) identificador. Para todas as variáveis específicas de serviço, consulte [the section called “Lista de identificadores específicos de serviço”](#).

O exemplo a seguir de variável de ambiente define um endpoint para o AWS Elastic Beanstalk:

Linux or macOS

```
$ export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

Windows Command Prompt

Como definir para todas as sessões

```
C:\> setx AWS_ENDPOINT_URL_ELASTIC_BEANSTALK http://localhost:4567
```

Como definir somente para a sessão atual

```
C:\> set AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL_ELASTIC_BEANSTALK="http://localhost:4567"
```

Para obter mais informações sobre como definir variáveis de ambiente, consulte [the section called “Variáveis de ambiente”](#).

Endpoints específicos de serviço: arquivo compartilhado **config**

No arquivo compartilhado `config`, `endpoint_url` é usado em várias seções. Para definir um endpoint específico de serviço, use a configuração `endpoint_url` aninhada em uma chave de identificação de serviço em uma seção `services`. Para obter detalhes sobre como definir uma seção `services` no arquivo compartilhado [the section called “services”](#), consulte `config`.

O exemplo a seguir usa uma `services` seção para configurar um endpoint específico de serviço para o Amazon URL S3 e um endpoint global personalizado usado para todos os outros serviços:

```
[profile dev1]  
endpoint_url = http://localhost:1234  
services = s3-specific  
  
[services testing-s3]  
s3 =  
  endpoint_url = http://localhost:4567
```

Um único perfil pode configurar endpoints para vários serviços. O exemplo a seguir define o endpoint específico do serviço para o Amazon URLs S3 e AWS Elastic Beanstalk no mesmo perfil.

Para obter uma lista de todas as chaves de identificação de serviço a serem usadas na seção `services`, consulte [Lista de identificadores específicos de serviço](#).

```
[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
```

A seção de configuração de serviço pode ser usada em vários perfis. O exemplo a seguir tem dois perfis que usam a mesma definição de `services`:

```
[profile dev1]
output = json
services = testing-s3

[profile dev2]
output = text
services = testing-s3

[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567
```

Endpoints específicos de serviço: lista de identificadores específicos de serviço

O AWS service (Serviço da AWS) identificador é baseado no API modelo, substituindo todos os espaços `serviceId` por sublinhados e colocando todas as letras em minúsculas.

O exemplo de identificador de serviço a seguir usa AWS Elastic Beanstalk. AWS Elastic Beanstalk tem um `serviceId` de [Elastic Beanstalk](#), portanto, a chave do identificador de serviço é `elastic_beanstalk`.

A tabela a seguir lista todos os identificadores, chaves do arquivo `config` e variáveis de ambiente específicos de serviço.

Precedência de configurações e definições do endpoint

As configurações do endpoint estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. As configurações do endpoint da AWS CLI têm precedência na seguinte ordem:

1. A opção da linha de comando `--endpoint-url`.
2. Se habilitada, a variável de ambiente global `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` ou a configuração do perfil `ignore_configure_endpoint_urls` para ignorar os endpoints personalizados.
3. O valor fornecido por uma variável de ambiente específica do serviço `AWS_ENDPOINT_URL_<SERVICE>`, como `AWS_ENDPOINT_URL_DYNAMODB`.
4. Os valores fornecidos pelas variáveis de ambiente `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT` e `AWS_ENDPOINT_URL`.
5. O valor do endpoint específico de serviço fornecido pela configuração `endpoint_url` em uma seção `services` do arquivo compartilhado `config`.
6. O valor fornecido pela configuração `endpoint_url` em um `profile` do arquivo compartilhado `config`.
7. Configurações `use_dualstack_endpoint`, `use_fips_endpoint` e `endpoint_url`.
8. Qualquer endpoint padrão URL para o respectivo AWS service (Serviço da AWS) é usado por último. Para obter uma lista dos endpoints de serviços padrão disponíveis em cada região, consulte [Regiões e endpoints da AWS](#) no Referência geral da Amazon Web Services.

Autenticação e credenciais de acesso

Você deve estabelecer como eles AWS CLI se autenticam AWS quando você desenvolve com AWS serviços. Para configurar credenciais para acesso programático ao AWS CLI, escolha uma das opções a seguir. As opções estão em ordem de recomendação.

Qual usuário precisa de acesso programático?	Finalidade	Instruções
IAM	Use credenciais de curto prazo.	the section called “Credenciais de curto prazo”
IAM	Use perfis como credenciais.	the section called “IAMfunções”
IAM	(Não recomendado) Use credenciais de longo prazo.	the section called “IAMusuários”

Precedência de credenciais e configurações

As credenciais e as definições de configuração estão localizadas em vários locais, como as variáveis de ambiente do sistema ou do usuário, arquivos de AWS configuração locais ou declaradas explicitamente na linha de comando como um parâmetro. Certos locais têm precedência sobre outros. As credenciais da AWS CLI e as definições de configuração têm precedência na seguinte ordem:

1. [Opções da linha de comando](#): substituem as configurações em qualquer outro local, como nos parâmetros `--region`, `--output` e `--profile`.
2. [Variáveis de ambiente](#): você pode armazenar valores nas variáveis de ambiente do sistema.
3. [Assumir perfil](#): assumo as permissões de um perfil do IAM por meio da configuração ou do comando `aws sts assume-role`.
4. [Assumir perfil com identidade da web](#): assumo as permissões de um perfil do IAM usando uma identidade da web por meio da configuração ou do comando `aws sts assume-role`.
5. [Arquivo de credenciais](#): os arquivos `credentials` e `config` são atualizados quando você executa o comando `aws configure`. O arquivo `credentials` está localizado em `~/.aws/`

credentials no Linux ou MacOS ou em C:\Users*USERNAME*\.aws\credentials no Windows.

6. [Processo personalizado](#): obtenha suas credenciais de uma fonte externa.
7. [Arquivo de configuração](#): os arquivos credentials e config são atualizados quando você executa o comando `aws configure`. O arquivo config está localizado em `~/.aws/config` no Linux ou MacOS ou em `C:\Users\USERNAME\.aws\config` no Windows.
8. [Credenciais de container](#): você pode associar uma função do IAM a cada uma das suas definições de tarefa do Amazon Elastic Container Service (Amazon ECS). As credenciais temporárias para essa função estão disponíveis para os contêineres dessa tarefa. Para mais informações, consulte [Funções do IAM para Tarefas](#) no Guia de Desenvolvedor Amazon Elastic Container Service.
9. [Credenciais de perfil de instância](#): você pode associar um perfil do IAM a cada uma das suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2). As credenciais temporárias para essa função estão disponíveis para o código em execução na instância. As credenciais são fornecidas por meio do serviço de metadados do Amazon EC2. Para obter mais informações, consulte [Funções do IAM para o Amazon EC2](#) no Guia do usuário do Amazon EC2 e [Como usar perfis de instância](#) no Guia do usuário do IAM.

Tópicos adicionais nesta seção

- [the section called “Credenciais de curto prazo”](#)
- [the section called “IAMfunções”](#)
- [the section called “IAMusuários”](#)
- [the section called “Use metadados de EC2 instâncias da Amazon para credenciais AWS CLI”](#)
- [the section called “Credenciais externas”](#)

Autenticar com credenciais de curto prazo

Recomendamos configurar sua ferramenta SDK ou para usar a [autenticação do IAM Identity Center](#) com opções de duração de sessão estendida. No entanto, você pode copiar e usar credenciais temporárias que estão disponíveis no portal de AWS acesso. As novas credenciais precisarão ser copiadas quando essas expirarem. É possível usar as credenciais temporárias em um perfil ou usá-las como valores para propriedades do sistema e variáveis de ambiente.

1. [Faça login no portal de AWS acesso.](#)

Use uma IAM função no AWS CLI

Uma [função AWS Identity and Access Management \(IAM\)](#) é uma ferramenta de autorização que permite ao usuário obter permissões adicionais (ou diferentes) ou obter permissões para realizar ações em uma AWS conta diferente.

Tópicos

- [Pré-requisitos](#)
- [Visão geral do uso de IAM funções](#)
- [Configurar e usar uma função](#)
- [Usar autenticação multifator](#)
- [Funções entre contas e ID externo](#)
- [Especificar um nome de sessão de função para facilitar a auditoria](#)
- [Assumir a função com a identidade da web](#)
- [Limpar as credenciais em cache](#)

Pré-requisitos

Para executar os comandos `iam`, você precisa instalar e configurar a AWS CLI. Para ter mais informações, consulte [Instalar a AWS CLI](#).

Visão geral do uso de IAM funções

Você pode configurar o AWS Command Line Interface (AWS CLI) para usar uma IAM função definindo um perfil para a função no `~/.aws/config` arquivo.

O exemplo a seguir mostra um perfil de função chamado `marketingadmin`. Se você executar comandos com `--profile marketingadmin` (ou especificá-los com a [variável de PROFILE ambiente AWS_](#)), eles AWS CLI usarão as credenciais definidas em um perfil separado `user1` para assumir a função com o Amazon Resource Name (ARN) `arn:aws:iam::123456789012:role/marketingadminrole`. É possível executar quaisquer operações que forem permitidas pelas permissões atribuídas a essa função.

```
[profile marketingadmin]  
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
```

```
source_profile = user1
```

Então, é possível especificar um `source_profile` que aponte para um perfil nomeado separado que contenha credenciais de usuário com permissão para usar o perfil. No exemplo anterior, o perfil `marketingadmin` usa as credenciais no perfil `user1`. Quando você especifica que um AWS CLI comando deve usar o perfil `marketingadmin`, ele pesquisa AWS CLI automaticamente as credenciais do `user1` perfil vinculado e as usa para solicitar credenciais temporárias para a função especificada IAM. CLI Ele usa a `AssumeRole` operação [sts](#): em segundo plano para fazer isso. Essas credenciais temporárias são então usadas para executar o comando da AWS CLI solicitado. A função especificada deve ter políticas de IAM permissão anexadas que permitam a execução do AWS CLI comando solicitado.

Para executar um AWS CLI comando de dentro de uma instância do Amazon Elastic Compute Cloud (AmazonEC2) ou de um contêiner do Amazon Elastic Container Service (AmazonECS), você pode usar uma IAM função anexada ao perfil da instância ou ao contêiner. Se você não especificar nenhum perfil ou não definir variáveis de ambiente, essa função será usada diretamente. Isso permite que você evite armazenar chaves de acesso de longa duração em suas instâncias. Também é possível usar essas funções de instância ou contêiner apenas para obter credenciais para outra função. Para isso, use `credential_source` (em vez de `source_profile`) para especificar como localizar as credenciais. O atributo `credential_source` oferece suporte aos seguintes valores:

- `Environment`: recupera as credenciais de origem de variáveis de ambiente.
- `Ec2InstanceMetadata`— Usa a IAM função anexada ao perfil da EC2 instância da Amazon.
- `EcsContainer`— Usa a IAM função anexada ao ECS contêiner da Amazon.

O exemplo a seguir mostra a mesma `marketingadminrole` função usada ao referenciar um perfil de EC2 instância da Amazon.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
```

Ao invocar uma função, você tem opções adicionais que podem ser exigidas, como o uso de autenticação multifator e um ID externo (usado por empresas de terceiros para acessar os recursos de seus clientes). Você também pode especificar nomes de sessão de função exclusivos que podem ser auditados com mais facilidade nos AWS CloudTrail registros.

Configurar e usar uma função

Quando você executa comandos usando um perfil que especifica uma IAM função, ele AWS CLI usa as credenciais do perfil de origem para chamar AWS Security Token Service (AWS STS) e solicitar credenciais temporárias para a função especificada. O usuário no perfil de origem deve ter permissão para chamar `sts:assume-role` para a função no perfil especificado. A função deve ter uma relação de confiança que permita que o usuário no perfil de origem use a função. Geralmente, o processo de recuperar e depois usar credenciais temporárias para uma função é chamado de assumir a função.

Você pode criar uma função IAM com as permissões que deseja que os usuários assumam seguindo o procedimento em [Criação de uma função para delegar permissões a um IAM usuário](#) no Guia do AWS Identity and Access Management usuário. Se a função e o usuário do do perfil de origem estão na mesma conta, é possível inserir seu próprio ID de conta ao configurar a relação de confiança da função.

Depois de criar a função, modifique a relação de confiança para permitir que o usuário assumir .

O exemplo a seguir mostra uma política de confiança que pode ser associada a uma função. Essa política permite que o perfil seja assumido por qualquer usuário na conta 123456789012, se o administrador dessa conta conceder explicitamente a permissão `sts:AssumeRole` ao usuário.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

A política de confiança não concede permissões. O administrador da conta deve delegar a permissão para assumir a função para usuários individuais, anexando uma política com as permissões apropriadas. O exemplo a seguir mostra uma política que pode ser associada a um usuário, permitindo que ele assuma apenas o perfil `marketingadminrole`. Para obter mais informações

sobre como conceder a um usuário acesso para assumir uma função, consulte [Conceder permissão ao usuário para trocar de função no Guia](#) do IAMusuário.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

O usuário não precisa ter permissões adicionais para executar os AWS CLI comandos usando o perfil da função. Em vez disso, as permissões para executar o comando vêm das associadas à função. Você anexa políticas de permissão à função para especificar quais ações podem ser executadas em quais AWS recursos. Para obter mais informações sobre como anexar permissões a uma função (que funciona de forma idêntica à de um usuário), consulte [Alteração de permissões para um IAM usuário](#) no Guia do IAMusuário.

Agora que o perfil de função, as permissões de função, a relação de confiança de função e as permissões de usuário estão configurados corretamente, é possível usar a função na linha de comando invocando a opção `--profile`. Por exemplo, o indicado a seguir chama o comando `ls` do Amazon S3 usando as permissões anexadas à função `marketingadmin`, conforme definido no exemplo no início desse tópico.

```
$ aws s3 ls --profile marketingadmin
```

Para usar a função para várias chamadas, defina a variável de ambiente `AWS_PROFILE` para a sessão atual na linha de comando. Enquanto essa variável de ambiente é definida, você não precisa especificar a opção `--profile` em cada comando.

Linux ou macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

Para obter mais informações sobre como configurar usuários e funções, consulte [IAMIdentities \(usuários, grupos de usuários e funções\)](#) e [IAMfunções no Guia](#) do IAMusuário.

Usar autenticação multifator

Para segurança adicional, você pode exigir que os usuários forneçam uma chave única gerada a partir de um dispositivo de autenticação multifator (MFA), um dispositivo U2F ou aplicativo móvel quando tentarem fazer uma chamada usando o perfil da função.

Primeiro, você pode optar por modificar a relação de confiança na IAM função a ser exigidaMFA. Isso impede que qualquer pessoa use a função sem primeiro se autenticar usandoMFA. Por exemplo, veja a linha `Condition` no exemplo a seguir. Essa política permite que o usuário nomeado assuma a função `anika` à qual a política está vinculada, mas somente se ele se autenticar usandoMFA.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/anika" },
      "Action": "sts:AssumeRole",
      "Condition": { "Bool": { "aws:multifactorAuthPresent": true } }
    }
  ]
}
```

Em seguida, adicione uma linha ao perfil da função que especifica o ARN MFA dispositivo do usuário. As entradas do exemplo de arquivo `config` a seguir mostram dois perfis que usam as chaves de acesso para o usuário `anika` solicitar credenciais temporárias para o perfil `cli-role`. O usuário `anika` tem permissão para assumir a função, concedida pela política de confiança da função.

```
[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
```

A `mfa_serial` configuração pode usar um ARN, conforme mostrado, ou o número de série de um MFA token de hardware.

O primeiro perfil `role-without-mfa`, não exige MFA. No entanto, como o exemplo anterior da política de confiança anexada à função exige MFA, qualquer tentativa de executar um comando com esse perfil falha.

```
$ aws iam list-users --profile role-without-mfa
```

```
An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

A segunda entrada do perfil, `role-with-mfa`, identifica um MFA dispositivo a ser usado. Quando o usuário tenta executar um AWS CLI comando com esse perfil, ele AWS CLI solicita que o usuário insira a senha de uso único (OTP) fornecida pelo MFA dispositivo. Se a MFA autenticação for bem-sucedida, o comando executará a operação solicitada. O não OTP é exibido na tela.

```
$ aws iam list-users --profile role-with-mfa
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
{
  "Users": [
    {
      ...
    }
  ]
}
```

Funções entre contas e ID externo

É possível permitir que os usuários do usem funções que pertençam a diferentes contas ao configurar a função como uma função entre contas. Durante a criação da função, defina o tipo de função como Outra AWS conta, conforme descrito em [Criação de uma função para delegar permissões a um IAM usuário](#). Opcionalmente, selecione Exigir MFA. Exigir MFA configura a condição apropriada na relação de confiança, conforme descrito em [Usar autenticação multifator](#).

Se usar um [ID externo](#) para fornecer mais controle sobre quem pode usar uma função em todas as contas, você também deverá adicionar o parâmetro `external_id` ao perfil da função. Normalmente, isso é usado somente quando a outra conta é controlada por alguém de fora da sua empresa ou organização.

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

Especificar um nome de sessão de função para facilitar a auditoria

Quando muitos indivíduos compartilham uma função, a auditoria torna-se um desafio. É recomendável associar cada operação invocada ao indivíduo que invocou a ação. No entanto, quando o indivíduo usa uma função, a assunção da função por ele é uma ação separada da invocação de uma operação, e é necessário correlacionar manualmente as duas.

É possível simplificar isso especificando nomes de sessão de função exclusivos quando os usuários assumem uma função. Para fazer isso, adicione um parâmetro `role_session_name` a cada perfil nomeado no arquivo `config` que especifica uma função. O `role_session_name` valor é passado para a `AssumeRole` operação e se torna parte da ARN sessão da função. Também está incluído nos AWS CloudTrail registros de todas as operações registradas.

Por exemplo, é possível criar um perfil baseado em função da maneira a seguir.

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
role_session_name = Session_Maria_Garcia
```

Isso faz com que a sessão de função tenha o seguinte ARN.

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

Além disso, todos os AWS CloudTrail registros incluem o nome da sessão da função nas informações capturadas para cada operação.

Assumir a função com a identidade da web

Você pode configurar um perfil para indicar que eles AWS CLI devem assumir uma função usando a [federação de identidade da web e o Open ID Connect \(OIDC\)](#). Quando você especifica isso em um perfil, ele faz AWS CLI automaticamente a AWS STS `AssumeRoleWithWebIdentity` chamada correspondente para você.

Note

Quando você especifica um perfil que usa uma IAM função, ele AWS CLI faz as chamadas apropriadas para recuperar credenciais temporárias. Essas credenciais são armazenadas em `~/.aws/cli/cache`. AWS CLI Os comandos subsequentes que especificam o mesmo perfil usam as credenciais temporárias em cache até que elas expirem. Nesse ponto, ele atualiza AWS CLI automaticamente as credenciais.

Para recuperar e usar credenciais temporárias usando a federação de identidades da web, é possível especificar os valores de configuração a seguir em um perfil compartilhado.

[role_arn](#)

Especifica a função ARN a ser assumida.

[web_identity_token_file](#)

Especifica o caminho para um arquivo que contém um token de acesso OAuth 2.0 ou token de ID do OpenID Connect fornecido pelo provedor de identidade. A AWS CLI carrega esse arquivo e transmite seu conteúdo como o argumento `WebIdentityToken` da operação `AssumeRoleWithWebIdentity`.

[role_session_name](#)

Especifica um nome opcional aplicado a essa sessão `assume-role`.

Veja a seguir um exemplo da configuração mínima necessária para uma função de admissão com o perfil de identidade da web:

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoleNameToAssume
```

```
web_identity_token_file=/path/to/a/token
```

Também é possível fornecer essa configuração usando [variáveis de ambiente](#).

AWS_ROLE_ARN

O ARN do papel a ser assumido.

AWS_WEB_IDENTITY_TOKEN_FILE

O caminho para o arquivo de token de identidade da web.

AWS_ROLE_SESSION_NAME

O nome aplicado a essa sessão assume-role.

Note

No momento, essas variáveis de ambiente se aplicam somente à função de admissão com o provedor de identidade da web. Elas não se aplicam à configuração geral do provedor de função.

Limpar as credenciais em cache

Quando você usa uma função, ele armazena em AWS CLI cache as credenciais temporárias localmente até que elas expirem. Na próxima vez que você tentar usá-los, eles AWS CLI tentarão renová-los em seu nome.

Se as credenciais temporárias da função forem [revogadas](#), elas não serão renovadas automaticamente, e as tentativas de usá-las falharão. No entanto, você pode excluir o cache para forçar o AWS CLI a recuperar novas credenciais.

Linux ou macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\aws\cli\cache
```

Autenticar com credenciais IAM de usuário

Warning

Para evitar riscos de segurança, não use IAM usuários para autenticação ao desenvolver software específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como [AWS IAM Identity Center](#).

Esta seção explica como definir as configurações básicas com um IAM usuário. Isso inclui suas credenciais de segurança usando os arquivos `config` e `credentials`.

Tópicos

- [Etapa 1: Crie seu IAM usuário](#)
- [Etapa 2: obter as chaves de acesso](#)
- [Configure o AWS CLI](#)
 - [Utilizar o `aws configure`](#)

Etapa 1: Crie seu IAM usuário

Crie seu IAM usuário seguindo o procedimento [Criação de IAM usuários \(console\)](#) no Guia do IAM usuário.

- Em Opções de permissão, escolha Anexar políticas diretamente para como você deseja atribuir permissões a esse usuário.
- A maioria dos SDK tutoriais de “Introdução” usa o serviço Amazon S3 como exemplo. Para fornecer à aplicação acesso total ao Amazon S3, selecione a política `AmazonS3FullAccess` para anexar a esse usuário.

Etapa 2: obter as chaves de acesso

1. Faça login no AWS Management Console e abra o IAM console em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do IAM console, selecione Usuários e, em seguida, selecione o **User name** usuário que você criou anteriormente.

3. Na página do usuário, selecione a página Credenciais de segurança. Depois, em Chaves de acesso, selecione Criar chave de acesso.
4. Para Criar chave de acesso Etapa 1, escolha Interface de linha de comando (CLI).
5. Em Criar chave de acesso: etapa 2, insira uma tag opcional e selecione Próximo.
6. Em Criar chave de acesso, Etapa 3, selecione Baixar arquivo.csv para salvar um **.csv** arquivo com a chave de acesso e a chave de acesso secreta do IAM usuário. Você precisará dessas informações posteriormente.
7. Selecione Concluído.

Configure o AWS CLI

Para uso geral, AWS CLI ele precisa das seguintes informações:

- Access key ID (ID da chave de acesso)
- Secret access key (Chave de acesso secreta)
- AWS Região
- Formato de saída

O AWS CLI armazena essas informações em um perfil (uma coleção de configurações) nomeado `default` no `credentials` arquivo. Por padrão, as informações desse perfil são usadas quando você executa um AWS CLI comando que não especifica explicitamente um perfil a ser usado. Para obter mais informações sobre o arquivo `credentials`, consulte [Configurações do arquivo de configuração e credenciais](#).

Para configurar o AWS CLI, use um dos procedimentos a seguir:

Tópicos

- [Utilizar o `aws configure`](#)

Utilizar o `aws configure`

Para uso geral, o `aws configure` comando é a maneira mais rápida de configurar sua AWS CLI instalação. Esse assistente de configuração solicita cada informação necessária para começar. A menos que especificado de outra forma usando a `--profile` opção, o AWS CLI armazena essas informações no `default` perfil.

O exemplo a seguir configura um perfil default usando valores de amostra. Substitua-os por seus próprios valores, conforme descrito nas seções a seguir.

```
$ 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
```

O exemplo a seguir configura um perfil chamado userprod usando valores de amostra. Substitua-os por seus próprios valores, conforme descrito nas seções a seguir.

```
$ 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
```

Use metadados de EC2 instâncias da Amazon para credenciais AWS CLI

Ao executar o AWS CLI de dentro de uma instância do Amazon Elastic Compute Cloud (AmazonEC2), você pode simplificar o fornecimento de credenciais para seus comandos. Cada EC2 instância da Amazon contém metadados que AWS CLI podem ser consultados diretamente em busca de credenciais temporárias. Quando uma IAM função é anexada à instância, ela recupera AWS CLI automaticamente e com segurança as credenciais dos metadados da instância.

Para desativar esse serviço, use a variável de DISABLED ambiente [AWSEC2METADATA_](#).

Tópicos

- [Pré-requisitos](#)
- [Configurando um perfil para metadados da Amazon EC2](#)

Pré-requisitos

Para usar EC2 as credenciais da Amazon com o AWS CLI, você precisa concluir o seguinte:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI e Autenticação e credenciais de acesso](#).
- Conhecer os arquivos de configuração e os perfis nomeados. Para ter mais informações, consulte [Configurações do arquivo de configuração e credenciais](#).
- Você criou uma função AWS Identity and Access Management (IAM) que tem acesso aos recursos necessários e anexou essa função à EC2 instância da Amazon ao iniciá-la. Para obter mais informações, consulte [IAMas políticas da Amazon EC2](#) no Guia EC2 do usuário da Amazon e [Conceder acesso a AWS recursos para aplicativos executados em EC2 instâncias da Amazon](#) no Guia do IAM usuário.

Configurando um perfil para metadados da Amazon EC2

Para especificar que você deseja usar as credenciais disponíveis no perfil de EC2 instância de hospedagem da Amazon, use a seguinte sintaxe no perfil nomeado em seu arquivo de configuração. Veja as etapas a seguir para obter mais instruções.

```
[profile profilename]  
role_arn = arn:aws:iam::123456789012:role/rolename  
credential_source = Ec2InstanceMetadata  
region = region
```

1. Crie um perfil em seu arquivo de configuração.

```
[profile profilename]
```

2. Adicione sua função IAM arn que tenha acesso aos recursos necessários.

```
role_arn = arn:aws:iam::123456789012:role/rolename
```

3. Especifique Ec2InstanceMetadata como sua fonte de credencial.

```
credential_source = Ec2InstanceMetadata
```

4. Defina sua região.

```
region = region
```

Exemplo

O exemplo a seguir pressupõe que *função de administrador de marketing* função e usa a *us-west-2* região em um perfil de EC2 instância da Amazon chamado *marketingadmin*.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
region = us-west-2
```

Credenciais de origem com um processo externo

Warning

Este tópico discute as credenciais de fornecimento de um processo externo. Isso poderá ser um risco de segurança se o comando para gerar as credenciais se tornar acessível por processos ou usuários não aprovados. Recomendamos que você use as alternativas seguras e suportadas fornecidas pelo AWS CLI e AWS para reduzir o risco de comprometer suas credenciais. Certifique-se de proteger o arquivo `config` e todos os arquivos e ferramentas com suporte para impedir a divulgação.

Certifique-se de que sua ferramenta de credencial personalizada não grave nenhuma informação secreta, `StdErr` pois ela AWS CLI pode capturar SDKs e registrar essas informações, potencialmente expondo-as a usuários não autorizados.

Se você tiver um método para gerar ou pesquisar credenciais que não seja diretamente suportado pelo AWS CLI, você pode configurar o AWS CLI para usá-lo definindo a `credential_process` configuração no `config` arquivo.

Por exemplo, você pode incluir uma entrada semelhante à seguinte no arquivo `config`.

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

Sintaxe

Para criar essa string de uma forma compatível com qualquer sistema operacional, siga estas regras:

- Se o caminho ou o nome do arquivo contiver um espaço, coloque o caminho completo e o nome do arquivo entre aspas duplas (" "). O caminho e o nome do arquivo podem ter somente os caracteres: A–Z a–z 0–9 - _ . espaço
- Se um nome de parâmetro ou um valor de parâmetro tiver um espaço, coloque esse elemento entre aspas duplas (" "). Coloque somente o nome ou o valor entre aspas, não o par.
- Não inclua variáveis de ambiente nas strings. Por exemplo, não inclua \$HOME ou %USERPROFILE %.
- Não especifique a pasta base como ~. É necessário especificar o caminho completo.

Exemplo para Windows

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter with spaces"
```

Exemplo para Linux ou macOS

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces "parameter with spaces"
```

Saída esperada do programa de credenciais


O AWS CLI executa o comando conforme especificado no perfil e, em seguida, lê os dados de STDOUT. O comando especificado deve gerar uma JSON saída STDOUT que corresponda à sintaxe a seguir.

```
{
  "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

No momento da elaboração deste documento, a chave `Version` deve ser definida como `1`. Isso pode aumentar ao longo do tempo conforme a estrutura evolui.

A `Expiration` chave é um carimbo de data/hora formatado em [ISO8601](#). Se a `Expiration` chave não estiver presente na saída da ferramenta, CLI presume-se que as credenciais são credenciais de longo prazo que não são atualizadas. Caso contrário, as credenciais serão consideradas temporárias e serão atualizadas automaticamente com a nova execução do comando `credential_process` antes de expirarem.

 Note

O AWS CLI não armazena em cache as credenciais do processo externo da mesma forma que assume as credenciais de função. Se o armazenamento em cache for obrigatório, implemente-o no processo externo.

O processo externo pode retornar um código de retorno diferente de zero para indicar que ocorreu um erro ao recuperar as credenciais.

Use o AWS CLI

Esta seção fornece uma visão geral abrangente do uso geral, dos recursos comuns e das opções disponíveis no AWS Command Line Interface (AWS CLI), indo além dos detalhes abordados na [the section called “Endpoints”](#) seção Configuração.

Este guia aborda os aspectos fundamentais da escrita de AWS CLI comandos, incluindo sua estrutura básica, formatação e recursos de filtragem. Ao entender esses elementos principais, você poderá criar comandos que direcionam com precisão os recursos e as ações de que você precisa, sem a necessidade de navegar em consoles complexos baseados na web.

Além disso, isso destaca o conteúdo de ajuda e a documentação disponíveis para AWS CLI o. Da ajuda integrada da linha de comando ao [guia de AWS CLI referência](#), você terá acesso a informações para ajudá-lo a explorar os recursos e capacidades do AWS CLI.

Para exemplos AWS service (Serviço da AWS) específicos e casos de uso, consulte o guia [Exemplos de código](#) de [AWS CLI referência da AWS CLI versão 2 do guia](#). Eles fornecem informações específicas sobre comandos e demonstram exemplos de como aproveitar o AWS CLI para vários Serviços da AWS.

Note

Por padrão, AWS CLI envia solicitações para Serviços da AWS usando HTTPS na TCP porta 443. Para garantir o uso bem-sucedido do AWS CLI, você deve ser capaz de fazer conexões de saída nessa porta.

Tópicos neste guia

- [Acesse a ajuda para o AWS Command Line Interface](#)
- [Estrutura de comando no AWS CLI](#)
- [Especifique os valores dos parâmetros para o AWS CLI](#)
- [Saída do comando de controle do AWS CLI](#)
- [Códigos de retorno da AWS CLI](#)
- [Crie e use atalhos de AWS CLI comando chamados aliases](#)

Acesse a ajuda para o AWS Command Line Interface

Este tópico descreve como acessar o conteúdo de ajuda para o AWS Command Line Interface (AWS CLI).

Tópicos

- [O comando de ajuda integrado da AWS CLI](#)
- [AWS CLI guia de referência](#)
- [API documentação](#)
- [Solucionar de problemas de erros](#)
- [Ajuda adicional](#)

O comando de ajuda integrado da AWS CLI

Você pode obter ajuda com qualquer comando ao usar o AWS Command Line Interface (AWS CLI). Para fazer isso, basta digitar `help` no final de um nome de comando.

Por exemplo, o comando a seguir exibe ajuda para as AWS CLI opções gerais e os comandos de nível superior disponíveis.

```
$ aws help
```

O comando a seguir exibe os comandos específicos disponíveis do Amazon Elastic Compute Cloud (AmazonEC2).

```
$ aws ec2 help
```

O exemplo a seguir mostra ajuda detalhada para a EC2 `DescribeInstances` operação da Amazon. A ajuda inclui descrições de seus parâmetros de entrada, filtros disponíveis, e o que é incluído como saída. Ela também inclui exemplos que mostram como digitar as variações comuns do comando.

```
$ aws ec2 describe-instances help
```

A ajuda para cada comando é dividida em seis seções:

Nome

O nome do comando.

```
NAME
    describe-instances -
```

Descrição

Uma descrição da API operação que o comando invoca.

```
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.

    ...
```

Resumo

A sintaxe básica para usar o comando e suas opções. Se uma opção estiver entre colchetes, significa que ela é opcional, tem um valor padrão ou tem uma opção alternativa que você pode usar.

```
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]
```

Por exemplo, `describe-instances` tem um comportamento padrão que descreve todas as instâncias na conta atual e AWS na região. Se preferir, você poderá especificar uma lista

de `instance-ids` para descrever uma ou mais instâncias. O `dry-run` é um sinalizador booleano opcional que não tem um valor. Para usar um sinalizador booleano, especifique um valor apresentado, nesse caso `--dry-run` ou `--no-dry-run`. Da mesma forma, `--generate-cli-skeleton` não tem um valor. Se houver condições no uso de uma opção, elas serão descritas na seção `OPTIONS` ou mostradas nos exemplos.

Opções

A descrição de cada uma das opções mostradas na sinopse.

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.

...

Exemplos

Os exemplos que mostram o uso do comando e suas opções. Se nenhum exemplo estiver disponível para um comando ou caso de uso necessário, solicite um usando o link de feedback nesta página ou na referência de AWS CLI comando na página de ajuda do 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"
```

```
...
```

Saída

As descrições de cada um dos campos e os tipos de dados incluídos na resposta da AWS.

Para `describe-instances`, a saída é uma lista de objetos de reserva, cada um com vários campos e objetos que contêm informações sobre as instâncias associadas a ele. Essas informações vêm da [API documentação do tipo de dados de reserva](#) usado pela 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 renderiza a saída emJSON, ela se torna uma matriz de objetos de reserva, semelhante ao exemplo a seguir.

```

{
  "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"
          }
        }
      ]
    }
  ]
}
...

```

Cada objeto de reserva contém campos que descrevem a reserva e uma série de objetos de instância, cada um com seus próprios campos (por exemplo, `PublicDnsName`) e objetos (por exemplo, `State`) que os descrevem.

Usuários do Windows

Você pode usar pipe (`|`) na saída do comando de ajuda do comando `more` para visualizar o arquivo de ajuda uma página por vez. Pressione a barra de espaço ou `PgDn` para ver mais do documento e `q` sair.

```
C:\> aws ec2 describe-instances help | more
```

AWS CLI guia de referência

Os arquivos de ajuda contêm links que não podem ser visualizados a partir da linha de comando nem é possível navegar até eles a partir dela. Você pode visualizar e interagir com esses links usando o guia de [referência on-line da AWS CLI versão 1, guia](#) de . A referência também contém o conteúdo de ajuda para todos os AWS CLI comandos. As descrições são apresentadas para facilitar a navegação e a visualização em dispositivos móveis, tablets e telas de desktop.

API documentação

Todos os comandos no AWS CLI correspondem às solicitações feitas ao público AWS de um serviço API. Cada serviço com um público API tem uma API referência que pode ser encontrada na página inicial do serviço no [site de AWS documentação](#). O conteúdo de uma API referência varia de acordo com a forma como API ela é construída e qual protocolo é usado. Normalmente, uma API referência contém informações detalhadas sobre as operações suportadas pelo API, os dados enviados de e para o serviço e quaisquer condições de erro que o serviço possa relatar.

API Seções de documentação

- **Ações:** informações detalhadas sobre cada operação e seus parâmetros (incluindo restrições de tamanho ou conteúdo e valores padrão). Ela lista os erros que podem ocorrer nessa operação. Cada operação corresponde a um subcomando no AWS CLI.
- **Tipos de dados:** informações detalhadas sobre estruturas que um comando pode exigir como um parâmetro ou retornar em resposta a uma solicitação.

- **Parâmetros comuns:** informações detalhadas sobre os parâmetros que são compartilhados por toda a ação para o serviço.
- **Erros comuns:** informações detalhadas sobre erros que podem ser retornados por todas as operações de um serviço.

O nome e a disponibilidade de cada seção podem variar de acordo com o serviço.

Específico do serviço CLIs

Alguns serviços têm uma separação CLI que data de antes da criação de um único AWS CLI para funcionar com todos os serviços. Esses serviços específicos CLIs têm documentação separada vinculada à página de documentação do serviço. A documentação para serviços específicos CLIs não se aplica ao. AWS CLI

Solucionar de problemas de erros

Para obter ajuda para diagnosticar e corrigir AWS CLI erros, consulte [Solucionar erros](#).

Ajuda adicional

Para obter ajuda adicional com seus AWS CLI problemas, visite a [AWS CLI comunidade](#) em GitHub.

Estrutura de comando no AWS CLI

Este tópico aborda como o comando AWS Command Line Interface (AWS CLI) é estruturado e como usar comandos de espera.

Tópicos

- [Estrutura do comando](#)
- [Comandos de espera](#)

Estrutura do comando

O AWS CLI usa uma estrutura de várias partes na linha de comando que deve ser especificada nesta ordem:

1. A chamada básica para o programa aws.
2. O comando de nível superior, que normalmente corresponde a um AWS serviço suportado pelo AWS CLI.
3. O subcomando que especifica a operação a ser realizada.
4. AWS CLI Opções ou parâmetros gerais exigidos pela operação. Você pode especificá-los em qualquer ordem, desde que siga as três primeiras partes. Se um parâmetro exclusivo for especificado várias vezes, apenas o último valor se aplicará.

```
$ aws <command> <subcommand> [options and parameters]
```

Os parâmetros podem assumir vários tipos de valores de entrada, como números, cadeias de caracteres, listas, mapas e JSON estruturas. O que é compatível depende do comando e do subcomando que você especificar.

Exemplos

Amazon S3

O exemplo a seguir lista todos os seus buckets do Amazon S3.

```
$ aws s3 ls
2018-12-11 17:08:50 my-bucket
2018-12-14 14:55:44 my-bucket2
```

Para obter mais informações sobre os comandos do Amazon S3, consulte [aws s3](#) na Referência de comandos da AWS CLI .

AWS CloudFormation

O exemplo de [create-change-set](#) comando a seguir altera o nome da pilha cloudformation para *my-change-set*.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

Para obter mais informações sobre os AWS CloudFormation comandos, consulte [aws cloudformation](#) na Referência de AWS CLI Comandos.

Comandos de espera

Alguns AWS serviços têm `wait` comandos disponíveis. Qualquer comando que usa `aws wait` normalmente espera até que um comando seja concluído antes de passar para a próxima etapa. Isso é especialmente útil para comandos em várias partes ou scripts, pois você pode usar um comando `wait` para impedir o avanço para etapas subsequentes se o comando `wait` falhar.

O AWS CLI usa uma estrutura de várias partes na linha de comando para o `wait` comando que deve ser especificada nesta ordem:

1. A chamada básica para o programa `aws`.
2. O comando de nível superior, que normalmente corresponde a um AWS serviço suportado pelo AWS CLI.
3. O comando `wait`.
4. O subcomando que especifica a operação a ser realizada.
5. CLI Opções ou parâmetros gerais exigidos pela operação. Você pode especificá-los em qualquer ordem, desde que siga as três primeiras partes. Se um parâmetro exclusivo for especificado várias vezes, apenas o último valor se aplicará.

```
$ aws <command> wait <subcommand> [options and parameters]
```

Os parâmetros podem assumir vários tipos de valores de entrada, como números, cadeias de caracteres, listas, mapas e JSON estruturas. O que é compatível depende do comando e do subcomando que você especificar.

Note

Nem todo AWS serviço oferece suporte a `wait` comandos. Consulte o guia de [AWS CLI referência AWS CLI do guia](#) para ver se seu serviço oferece suporte a `wait` comandos.

Exemplos

AWS CloudFormation

Os exemplos de [wait change-set-create-complete](#) comando a seguir são pausados e retomados somente depois de confirmar que o *my-change-set* alteração definida no *my-stack* a pilha está pronta para ser executada.

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

Para obter mais informações sobre os comandos AWS CloudFormation wait, consulte [wait](#) na Referência de comandos da AWS CLI .

AWS CodeDeploy

Os exemplos de [wait deployment-successful](#) comando a seguir fazem uma pausa até o *d-A1B2C3111* a implantação é concluída com êxito.

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

Para obter mais informações sobre os comandos AWS CodeDeploy wait, consulte [wait](#) na Referência de comandos da AWS CLI .

Especifique os valores dos parâmetros para o AWS CLI

Muitos parâmetros usados no AWS Command Line Interface (AWS CLI) são valores simples de string ou numéricos, como o nome do par de chaves *my-key-pair* no exemplo de comando a seguir `aws ec2 create-key-pair`.

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

A formatação do comando pode variar entre os terminais. Por exemplo, a maioria dos terminais faz distinção entre letras maiúsculas e minúsculas, mas não é o caso do Powershell. Isso significa que os dois exemplos de comando a seguir produziram resultados diferentes para terminais com distinção entre letras maiúsculas e minúsculas, pois eles veem *MyFile*.txt* e *myfile*.txt* como parâmetros diferentes.

No entanto, PowerShell processaria essas solicitações da mesma forma que vê *MyFile*.txt* e com os *myfile*.txt* mesmos parâmetros. O exemplo de comando a seguir demonstra esses parâmetros usando o `aws s3 cp` comando:

```
$ aws s3 cp . s3://my-bucket/path --include "MyFile*.txt"
```

```
$ aws s3 cp . s3://my-bucket/path --include "myfile*.txt"
```

Para obter mais informações sobre PowerShell a insensibilidade entre maiúsculas e minúsculas, consulte [about_Case-Sensitivity](#) na documentação. PowerShell

Às vezes, é necessário usar aspas ou literais em strings que incluem caracteres especiais ou de espaço. As regras sobre essa formatação também podem variar entre os terminais. Para obter mais informações sobre como usar aspas em parâmetros complexos, consulte [Aspas e literais com cadeias de caracteres no AWS CLI](#).

Esses tópicos abordam as regras de formatação de terminal mais comuns. Se você estiver tendo problemas com o seu terminal reconhecendo seus valores de parâmetros, certifique-se de revisar os tópicos desta seção e também de verificar a documentação do seu terminal para ver as regras de sintaxe específicas.

Tópicos de parâmetros

- [Tipos comuns de parâmetros da AWS CLI](#)
- [Aspas e literais com cadeias de caracteres no AWS CLI](#)
- [Carregar os parâmetros da AWS CLI de um arquivo](#)
- [Esqueletos e arquivos de entrada da AWS CLI](#)
- [Usar sintaxe simplificada com a AWS CLI](#)

Tipos comuns de parâmetros da AWS CLI

Esta seção descreve alguns dos tipos de parâmetros comuns e o formato típico necessário.

Caso haja problemas com a formatação de um parâmetro para um comando específico, verifique a ajuda inserindo **help** após o nome do comando. A ajuda de cada subcomando inclui o nome e a descrição de uma opção. O tipo de parâmetro da opção está listado entre parênteses. Para obter mais informações sobre a exibição da ajuda, consulte [the section called “Obter ajuda”](#).

Os tipos de parâmetro incluem:

- [String](#)
- [a função de hora](#)
- [Listar](#)
- [Boolean](#)
- [Inteiro](#)

- [Binário/blob \(objeto grande binário\) e blob de streaming](#)
- [Mapa](#)
- [Documento](#)

String

Os parâmetros de string podem conter caracteres alfanuméricos, símbolos e espaço em branco no conjunto de caracteres [ASCII](#). Strings com espaço em branco devem ser incluídas entre aspas. Recomendamos que você não use símbolos nem espaços em branco diferentes do caractere de espaço padrão e observe as [regras de aspas](#) para evitar resultados inesperados.

Alguns parâmetros de sequência de caracteres pode aceitar dados binários a partir de um arquivo. Consulte [Arquivos binários](#) para ver um exemplo.

a função de hora

As marcas de tempo são formatadas de acordo com a norma [ISO 8601](#). São geralmente chamados de parâmetros “DateTime” ou “Date”.

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

Os formatos aceitos incluem:

- *AAAA-MM-DDThh:mm:ss.sssTZD (UTC)*, por exemplo, 2014-10-01T20:30:00.000Z
- *AAAA-MM-DDThh:mm:ss.sssTZD (com deslocamento)*, por exemplo, 2014-10-01T12:30:00.000-08:00
- *AAAA-MM-DD*, por exemplo, 2014-10-01
- Tempo do Unix em segundos, por exemplo, 1412195400. Isso às vezes é conhecido como [Tempo epoch Unix](#) e representa o número de segundos desde a meia-noite, 1º de janeiro de 1970 UTC.

Você pode definir o formato do carimbo de data/hora usando a configuração do arquivo [cli_timestamp_format](#).

Listar

Uma ou mais strings separadas por espaços. Se qualquer um dos itens de string contiver espaços, você deverá colocar esse item entre aspas. Observe as [regras de aspas](#) de seu terminal para evitar resultados inesperados

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

Boolean

Sinalizador binário que ativa ou desativa uma opção. Por exemplo, `ec2 describe-spot-price-history` tem um parâmetro booleano `--dry-run` que, quando especificado, valida a consulta com o serviço sem realmente executá-la.

```
$ aws ec2 describe-spot-price-history --dry-run
```

A saída indica se o comando foi bem formado. Esse comando também inclui uma versão `--no-dry-run` do parâmetro que você pode usar para indicar explicitamente que o comando deve ser executado normalmente. A inclusão não é necessária, pois esse é o comportamento padrão.

Inteiro

Um número inteiro não assinado.

```
$ aws ec2 describe-spot-price-history --max-items 5
```

Binário/blob (objeto grande binário) e blob de streaming

Na AWS CLI, é possível transmitir um valor binário como uma string diretamente na linha de comando. Há dois tipos de blobs:

- [Blob](#)
- [Blob de streaming](#)

Blob

Para transmitir um valor para um parâmetro com o tipo blob, é necessário especificar um caminho para um arquivo local que contenha os dados binários usando o prefixo `fileb://`. Arquivos referenciados usando o prefixo `fileb://` são sempre tratados como um binário bruto não codificado. O caminho especificado é interpretado como sendo relativo ao diretório de trabalho atual. Por exemplo, o parâmetro `--plaintext` para `aws kms encrypt` é um blob.

```
$ aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://data.bin
```

```
--plaintext file://ExamplePlaintextFile \  
--output text \  
--query CiphertextBlob | base64 \  
--decode > ExampleEncryptedFile
```

Blob de streaming

Blobs de streaming como `aws cloudsearchdomain upload-documents` não usam prefixos. Em vez disso, os parâmetros de blob de streaming são formatados usando o caminho direto do arquivo. O seguinte exemplo usa o caminho direto do arquivo `document-batch.json` para o comando `aws cloudsearchdomain upload-documents`:

```
$ aws cloudsearchdomain upload-documents \  
  --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \  
  --content-type application/json \  
  --documents document-batch.json
```

Mapa

Um conjunto de pares de chave-valor especificado em JSON ou com a [sintaxe abreviada](#) da CLI. O exemplo de JSON a seguir lê um item de uma tabela do Amazon DynamoDB chamada `my-table` com um parâmetro de mapa, `--key`. O parâmetro especifica a chave primária chamada `id` com um valor numérico de 1 em uma estrutura JSON aninhada.

Para uso mais avançado de JSON em uma linha de comando, considere usar um processador JSON de linha de comando, como `jq`, para criar strings JSON. Para obter mais informações sobre o `jq`, consulte o [repositório do jq](#) no GitHub.

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N": "1"}}'  
  
{  
  "Item": {  
    "name": {  
      "S": "John"  
    },  
    "id": {  
      "N": "1"  
    }  
  }  
}
```


Documento

Note

A [sintaxe simplificada](#) não é compatível com tipos de documento.

Os tipos de documento são usados para enviar dados sem a necessidade de incorporar JSON dentro de strings. O tipo de documento permite que os serviços forneçam esquemas arbitrários para você usar tipos de dados mais flexíveis.

Isso permite enviar dados JSON sem precisar usar caracteres escape nos valores. Por exemplo, em vez de usar a seguinte entrada JSON com escape:

```
{"document": "{\"key\":true}"}
```

Você pode usar o seguinte tipo de documento:

```
{"document": {"key": true}}
```

Valores válidos para tipos de documento

Devido à natureza flexível dos tipos de documento, existem vários tipos de valor válidos. Entre os valores válidos estão os seguintes:

String

```
--option "value"
```

Número

```
--option 123  
--option 123.456
```

Boolean

```
--option true
```

Nulo

```
--option null
```

Array

```
--option '["value1", "value2", "value3"]'  
--option '["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]'
```

Objeto

```
--option '{"key": "value"}'  
--option '{"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":  
["value3", "value4"], "key6": {"value5": "value6"}'}
```

Aspas e literais com cadeias de caracteres no AWS CLI

Há duas formas principais de usar aspas simples e duplas na AWS CLI.

- [Uso de aspas em torno de strings que contêm espaços em branco](#)
- [Uso de aspas dentro de strings](#)

Uso de aspas em torno de strings que contêm espaços em branco

Os nomes dos parâmetros e valores são separados por espaços na linha de comando. Se um valor de string contiver um espaço incorporado, você deverá colocar toda a string entre aspas para evitar que o interprete mal o AWS CLI espaço como um divisor entre o valor e o próximo nome do parâmetro. O tipo de aspa que você usa depende do sistema operacional AWS CLI em que você está executando.

Linux and macOS

Uso de aspas simples ' '

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

Para obter mais informações sobre como usar aspas, consulte a documentação do usuário para o seu shell preferido.

PowerShell

Aspas simples (recomendado)

As aspas simples ' ' são chamadas de strings `verbatim`. A string é passada para o comando exatamente como você a digita, o que significa que PowerShell as variáveis não passarão.

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

Aspas duplas

As aspas duplas " " são chamadas de string `expandable`. As variáveis podem ser passadas em strings expansíveis.

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Para obter mais informações sobre o uso de aspas, consulte [Sobre as regras de cotação](#) no Microsoft PowerShell Docs.

Windows command prompt

Uso de aspas duplas " ".

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Opcionalmente, você pode separar o nome de parâmetro do valor com um sinal de igual =, em vez de um espaço. Isso geralmente é necessário apenas se o valor do parâmetro começa com um hífen.

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

Uso de aspas dentro de strings

As strings podem conter aspas, e seu shell pode exigir aspas de escape para que funcionem corretamente. Um dos tipos comuns de valor de parâmetro é uma JSON string. Isso é complexo, pois inclui espaços e aspas duplas " " ao redor do nome e valor de cada elemento na JSON estrutura. A forma como você insere os parâmetros JSON formatados na linha de comando varia de acordo com o sistema operacional.

Para um JSON uso mais avançado na linha de comando, considere usar um JSON processador de linha de comando, por exemplo `jq`, para criar JSON cadeias de caracteres. Para obter mais informações sobre `jq`, consulte o [repositório jq](#) em GitHub

Linux and macOS

Para que o Linux e o macOS interpretem cadeias de caracteres literalmente, use aspas simples `' '` para delimitar a estrutura de JSON dados, como no exemplo a seguir. Você não precisa escapar das aspas duplas incorporadas na JSON string, pois elas estão sendo tratadas literalmente. Como o JSON está entre aspas simples, todas as aspas simples na string precisarão ser escapadas, isso geralmente é feito usando uma barra invertida antes das aspas simples. `\'`

```
$ aws ec2 run-instances \  
  --image-id ami-12345678 \  
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":  
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]'
```

Para obter mais informações sobre como usar aspas, consulte a documentação do usuário para o seu shell preferido.

PowerShell

Use aspas simples `' '` ou aspas duplas `" "`.

Aspas simples (recomendado)

As aspas simples `' '` são chamadas de strings verbatim. A string é passada para o comando exatamente como você a digita, o que significa que PowerShell as variáveis não passarão.

Como as estruturas de JSON dados incluem aspas duplas, sugerimos aspas simples `' '` para incluí-las. Se você usar aspas simples, não precisará escapar das aspas duplas incorporadas na JSON string. No entanto, você precisa escapar de cada aspa com uma crase ``` dentro da JSON estrutura.

```
PS C:\> aws ec2 run-instances `  
  --image-id ami-12345678 `  
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":  
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]'
```

Aspas duplas

As aspas duplas " " são chamadas de string expandable. As variáveis podem ser passadas em strings expansíveis.

Se você usar aspas duplas, não precisará escapar das aspas simples incorporadas na JSON string. No entanto, você precisa escapar de cada aspa dupla com uma crase ` dentro da JSON estrutura, como no exemplo a seguir.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings "[{ `"/dev/sdb `", `Ebs `":
  { `"/VolumeSize `":20, `DeleteOnTermination `":false, `VolumeType `": `standard `"}]"
```

Para obter mais informações sobre o uso de aspas, consulte [Sobre as regras de cotação](#) no Microsoft PowerShell Docs.

Warning

Antes de PowerShell enviar um comando para o AWS CLI, ele determina se seu comando é interpretado usando regras típicas PowerShell ou de CommandLineToArgvW aspas. Ao PowerShell processar o usoCommandLineToArgvW, você deve escapar de caracteres com uma barra invertida\.

Para obter mais informações sobre *CommandLineToArgvW* PowerShell, consulte [O que está acontecendo com o estranho tratamento de aspas e barras invertidas por CommandLineToArgv W](#) no Microsoft DevBlogs, [Todos citam argumentos da linha de comando da maneira errada no](#) blog do Microsoft Docs e a função [CommandLineToArgvW no](#) Microsoft Docs.

Aspas simples

As aspas simples ' ' são chamadas de strings verbatim. A string é passada para o comando exatamente como você a digita, o que significa que PowerShell as variáveis não passarão. Caracteres de escape com barra invertida \.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings '[{ \"/dev/sdb \", \Ebs \":
  { \"/VolumeSize \":20, \DeleteOnTermination \":false, \VolumeType \": \standard \"}]'
```

Aspas duplas

As aspas duplas " " são chamadas de string expandable. As variáveis podem ser passadas em strings expansíveis. Para cadeias de caracteres com aspas duplas, você

precisa escapar duas vezes usando `\\` para cada citação em vez de usar apenas uma crase. O backtick escapa da barra invertida e, em seguida, a barra invertida é usada como um caractere de escape para o 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 (recomendados)

Para ignorar as regras de PowerShell cotação para entrada de JSON dados, use Blobs para passar seus JSON dados diretamente para o AWS CLI Para obter mais informações sobre o Blobs, consulte [Blob](#).

Windows command prompt

O prompt de comando do Windows exige aspas duplas `" "` para delimitar a estrutura de JSON dados. Além disso, para evitar que o processador de comandos interprete mal as aspas duplas incorporadas noJSON, você também deve excluir (preceder com um `\\` caractere de barra invertida) cada aspa dupla na própria estrutura de JSON dados, como `" "` no exemplo a seguir.

```
C:\> aws ec2 run-instances ^
  --image-id ami-12345678 ^
  --block-device-mappings "[{\\"DeviceName\\":\\"/dev/sdb\\",\\"Ebs\\":
  {\\"VolumeSize\\":20,\\"DeleteOnTermination\\":false,\\"VolumeType\\":\\"standard\\"}}]"
```

Somente as aspas duplas mais externas não são de escape.

Carregar os parâmetros da AWS CLI de um arquivo

Alguns parâmetros esperam nomes de arquivos como argumentos, dos quais a AWS CLI carrega os dados. Outros parâmetros permitem que você especifique o valor do parâmetro como texto digitado na linha de comando ou lido de um arquivo. Independentemente se um arquivo for obrigatório ou opcional, você deverá codificá-lo corretamente para que a AWS CLI possa entendê-lo. A codificação do arquivo deve corresponder à localidade padrão do sistema de leitura. É possível determinar isso usando o método `locale.getpreferredencoding()` do Python.

Note

Por padrão, o Windows PowerShell gera texto como UTF-16, o que entra em conflito com a codificação UTF-8 usada por arquivos JSON e muitos sistemas Linux. Recomendamos que você use `-Encoding ascii` com os comandos `Out-File` do PowerShell para garantir que a AWS CLI possa ler o arquivo resultante.

Tópicos

- [Como carregar os parâmetros de um arquivo](#)
- [Arquivos binários](#)
- [Arquivos remotos](#)

Como carregar os parâmetros de um arquivo

Às vezes é conveniente carregar um valor de parâmetro de um arquivo, em vez de tentar digitá-lo como um valor de parâmetro de linha de comando, por exemplo, quando o parâmetro é uma string JSON complexa. Para especificar um arquivo que contém o valor, especifique um URL de arquivo no formato a seguir.

```
file:///complete/path/to/file
```

- Os dois primeiros caracteres de barra “/” fazem parte da especificação. Se o caminho exigido começar com “/”, o resultado será três caracteres de barra: `file:///folder/file`.
- O URL fornece o caminho para o arquivo que apresenta o conteúdo real do parâmetro.
- Ao usar arquivos com espaços ou caracteres especiais, siga as [regras de aspas e escape](#) para seu terminal.

Note

Esse comportamento é desativado automaticamente para parâmetros que já esperam um URL, como um parâmetro que identifica um URL de modelo do AWS CloudFormation. Também é possível desativar esse comportamento desabilitando a configuração [cli_follow_urlparam](#) em seu arquivo de configuração AWS CLI.

Os caminhos do arquivo no exemplo a seguir são interpretados como sendo relativos ao diretório de trabalho atual.

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"
```

A opção de prefixo `file://` é compatível com as expansões de estilo Unix, inclusive `~/`, `./` e `../`. No Windows, a expressão `~/` se expande para o seu diretório de usuário, armazenado na variável de ambiente `%USERPROFILE%`. Por exemplo, no Windows 10, haveria normalmente um diretório de usuário em `C:\Users\UserName\`.

Ainda é necessário inserir um caractere de escape nos documentos JSON que são incorporados como o valor de outro documento JSON.

```
$ 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\\\"}"
}
```


Arquivos binários

Para comandos que incluem dados binários como um parâmetro, especifique que os dados são conteúdo binário usando o prefixo `fileb://`. Comandos que aceitam dados binários incluem:

- **aws ec2 run-instances**: parâmetro `--user-data`.
- **aws s3api put-object**: parâmetro `--sse-customer-key`.
- **aws kms decrypt**: parâmetro `--ciphertext-blob`.

O exemplo a seguir gera uma chave binária AES de 256 bits usando uma ferramenta de linha de comando do Linux e a fornece ao Amazon S3 para criptografar o arquivo enviado no lado do servidor.

```
$ 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 my-bucket \
  --key test.txt \
  --body test.txt \
  --sse-customer-key fileb://sse.key \
  --sse-customer-algorithm AES256
{
  "SSECustomerKeyMD5": "iVg8oWa8sy714+FjtesrJg==",
  "SSECustomerAlgorithm": "AES256",
  "ETag": "\"a6118e84b76cf98bf04bbe14b6045c6c\""
}
```

Arquivos remotos

A AWS CLI também é compatível com parâmetros de carregamento de um arquivo hospedado na Internet com um URL `http://` ou `https://`. O exemplo a seguir faz referência a um arquivo armazenado em um bucket do Amazon S3. Isso permite que você acesse arquivos de parâmetros de qualquer computador, mas requer que o contêiner seja acessível publicamente.

```
$ aws ec2 run-instances \
  --image-id ami-12345678 \
  --block-device-mappings http://my-bucket.s3.amazonaws.com/filename.json
```

O exemplo anterior assume que o arquivo `filename.json` contém os seguintes dados JSON.

```
[
  {
    "DeviceName": "/dev/sdb",
    "Ebs": {
      "VolumeSize": 20,
      "DeleteOnTermination": false,
      "VolumeType": "standard"
    }
  }
]
```

Para outro exemplo com referência a um arquivo que contém parâmetros formatados pelo JSON, consulte [Anexar uma política IAM gerenciada a um usuário](#).

Esqueletos e arquivos de entrada da AWS CLI

A maioria dos comandos da AWS CLI aceita todas as entradas de parâmetros de um arquivo. Esses modelos podem ser gerados usando a opção `generate-cli-skeleton`.

Tópicos

- [Sobre esqueletos e arquivos de entrada da AWS CLI](#)
- [Gerar um esqueleto de comando](#)

Sobre esqueletos e arquivos de entrada da AWS CLI

A maioria dos comandos da AWS Command Line Interface (AWS CLI) aceita todas as entradas de parâmetros usando os parâmetros `--cli-input-json`.

Felizmente, esses mesmos comandos fornecem o parâmetro `--generate-cli-skeleton` para gerar um arquivo JSON com todos os parâmetros que você pode editar e preencher. Depois, você pode executar o comando com o parâmetro relevante `--cli-input-json` e apontar para o arquivo preenchido.

Important

Vários comandos da AWS CLI não são mapeados diretamente para as operações individuais de API da AWS, como os comandos [aws s3](#). Esses comandos não são compatíveis com os parâmetros `--generate-cli-skeleton` ou `--cli-input-json` descritos neste tópico.

Se você souber se um comando específico é compatível com esses parâmetros, execute o comando a seguir, substituindo os nomes do *serviço* e do *comando* pelos quais você está interessado.

```
$ aws service command help
```

A saída inclui uma seção *Synopsis* que mostra os parâmetros compatíveis com o comando especificado.

```
$ aws iam list-users help
...
SYNOPSIS
    list-users
    ...
    [--cli-input-json]
    ...
    [--generate-cli-skeleton <value>]
...
```

O parâmetro `--generate-cli-skeleton` faz com que o comando não seja executado, mas, em vez disso, gere e exibe um modelo de parâmetro que você pode personalizar e usar como entrada em um comando posterior. O modelo gerado inclui todos os parâmetros compatíveis com o comando.

O parâmetro `--generate-cli-skeleton` aceita um dos seguintes valores:

- `input`: o modelo gerado inclui todos os parâmetros de entrada formatados como JSON. Este é o valor padrão.
- `output`: o modelo gerado inclui todos os parâmetros de saída formatados como JSON.

Como a AWS CLI é essencialmente um “wrapper” em torno da API de serviço, o arquivo de esqueleto espera que você faça referência a todos os parâmetros pelo nome do parâmetro da API subjacente. Isso provavelmente é diferente do nome do parâmetro da AWS CLI. Por exemplo, um parâmetro da AWS CLI chamado `user-name` pode ser mapeado para o parâmetro da API do serviço da AWS chamado `UserName` (observe a capitalização alterada e o traço ausente). Recomendamos usar a opção `--generate-cli-skeleton` para gerar o modelo com os nomes de parâmetro “corretos” a fim de evitar erros. Você também pode fazer referência ao Guia de referência

da API do serviço para ver os nomes de parâmetro esperados. É possível excluir quaisquer parâmetros do modelo que não sejam necessários e para os quais você não deseja fornecer um valor.

Por exemplo, se você executar o seguinte comando, ele gerará o modelo de parâmetro para o comando `run-instances` do Amazon Elastic Compute Cloud (Amazon EC2).

JSON

O exemplo a seguir mostra como gerar um modelo formatado em JSON usando o valor padrão (`input`) para o `--generate-cli-skeleton` parâmetro.

```
$ 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
```

```
}
```

Gerar um esqueleto de comando

Para gerar e usar um arquivo de esqueleto de parâmetro

1. Execute o comando com o parâmetro `--generate-cli-skeleton` para produzir JSON e direcionar a saída a um arquivo para salvá-lo.

JSON

```
$ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
```

2. Abra o arquivo de esqueleto do parâmetro em seu editor de texto e remova os parâmetros que não são mais necessários. Por exemplo, você pode reduzir o modelo ao indicado a seguir. Verifique se o arquivo ainda é JSON válido depois de remover os elementos que não são necessários.

JSON

```
{
  "DryRun": true,
  "ImageId": "",
  "KeyName": "",
  "SecurityGroups": [
    ""
  ],
  "InstanceType": "",
  "Monitoring": {
    "Enabled": true
  }
}
```

Neste exemplo, deixamos o parâmetro `DryRun` definido como `true` para usar o recurso de simulação do Amazon EC2. Esse recurso permite que você teste com segurança o comando sem realmente criar ou modificar nenhum recursos.

3. Preencha os valores restantes com valores apropriados para seu cenário. Neste exemplo, fornecemos o tipo de instância, o nome da chave, o grupo de segurança e o identificador da

imagem de máquina da Amazon (AMI) que devem ser usados. Este exemplo assume a região padrão da AWS. A AMI `ami-dfc39aef` é uma imagem de 64 bits do hospedada na região `us-west-2`. Se você usar uma região diferente, será necessário [encontrar o ID da AMI correta para usar](#).

JSON

```
{
  "DryRun": true,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

4. Execute o comando com os parâmetros concluídos passando o arquivo do modelo concluído o parâmetro `--cli-input-json` usando o prefixo `file://`. A AWS CLI interpreta o caminho a ser associado ao diretório de trabalho atual, de forma que o exemplo a seguir que exibe somente o nome do arquivo sem nenhum caminho seja procurado pelo arquivo diretamente no diretório de trabalho atual.

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.
```

O erro de simulação indica que JSON está formado corretamente e os valores de parâmetro são válidos. Se outros problemas forem relatados na saída, corrija-os e repita a etapa anterior até que a mensagem “Request would have succeeded” seja exibida.

5. Agora você pode definir o parâmetro `DryRun` como `false` para desativar a simulação.

JSON

```
{
  "DryRun": false,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

6. Execute o comando, e `run-instances` realmente iniciará uma instância do Amazon EC2 e exibirá os detalhes gerados pelo início bem-sucedido. O formato da saída é controlado pelo parâmetro `--output`, separadamente do formato do modelo de parâmetro de entrada.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
```

```
{
  "OwnerId": "123456789012",
  "ReservationId": "r-d94a2b1",
  "Groups": [],
  "Instances": [
    ...
  ]
}
```

Usar sintaxe simplificada com a AWS CLI

A AWS Command Line Interface (AWS CLI) pode aceitar muitos de seus parâmetros de opção no formato JSON. No entanto, pode ser entediante inserir grandes estruturas ou listas de JSON na linha de comando. Para tornar isso mais fácil, a AWS CLI também oferece suporte a uma sintaxe abreviada mais simples que permite a representação de seus parâmetros de opção em vez de utilizar o formato JSON completo.

Tópicos

- [Parâmetros de estrutura](#)
- [Usar sintaxe simplificada com a AWS Command Line Interface](#)

Parâmetros de estrutura

A sintaxe abreviada no AWS CLI facilita a inserção de parâmetros simples de entrada pelos usuários (estruturas não aninhadas). O formato é uma lista de pares de chave/valor separados por vírgula. Use as regras de [aspas](#) e escape apropriadas para seu terminal, pois a sintaxe abreviada são strings.

Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

PowerShell

```
--option "key1=value1,key2=value2,key3=value3"
```

Ambos são equivalentes ao exemplo a seguir formatado em JSON.

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

Não pode haver nenhum espaço em branco entre cada par de chave/valor separado por vírgula. Aqui está um exemplo do comando `update-table` do Amazon DynamoDB com a opção `--provisioned-throughput` especificada no formato simplificado.

```
$ aws dynamodb update-table \  
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \  
  --table-name MyDDBTable
```

Isso é equivalente ao exemplo a seguir formatado em JSON.

```
$ aws dynamodb update-table \  
  --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \  
  --table-name MyDDBTable
```

Usar sintaxe simplificada com a AWS Command Line Interface

Você pode especificar os parâmetros de entrada em um formulário de lista de duas formas: JSON ou abreviada. A sintaxe abreviada da AWS CLI é projetada para facilitar a inserção de listas com número, sequência de caracteres, estruturas aninhadas ou não.

O formato básico é mostrada aqui, onde os valores na lista são separados por um único espaço.

```
--option value1 value2 value3
```

Isso é equivalente ao exemplo a seguir, formatado em JSON.

```
--option '[value1,value2,value3]'
```

Como mencionado anteriormente, é possível especificar uma lista de números, uma lista de strings ou uma lista de estruturas de dados não aninhadas em formato abreviado. Veja a seguir um exemplo do comando `stop-instances` do Amazon Elastic Compute Cloud (Amazon EC2), onde o parâmetro de entrada (lista de strings) para a opção `--instance-ids` é especificado no formato simplificado.

```
$ aws ec2 stop-instances \  
  --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

Isso é equivalente ao exemplo a seguir formatado em JSON.

```
$ aws ec2 stop-instances \  
  --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

A seguir está um exemplo do comando `create-tags` do Amazon EC2, que leva uma lista de estruturas não aninhadas para a opção `--tags`. A opção `--resources` especifica o ID da instância a ser marcada.

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2  
Key=My3rdTag,Value=Value3
```

Isso é equivalente ao exemplo a seguir, formatado em JSON. O parâmetro JSON é escrito em várias linhas para melhor leitura.

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags '[  
    {"Key": "My1stTag", "Value": "Value1"},  
    {"Key": "My2ndTag", "Value": "Value2"},  
    {"Key": "My3rdTag", "Value": "Value3"}  
  ]'
```

Saída do comando de controle do AWS CLI

Esta seção descreve as diferentes maneiras de controlar a saída da AWS Command Line Interface (AWS CLI). Personalizar a AWS CLI saída em seu terminal pode melhorar a legibilidade, simplificar a automação de scripts e facilitar a navegação em conjuntos de dados maiores.

O AWS CLI suporta vários [formatos de saída json](#), incluindo [textyaml](#). Alguns serviços têm [paginação](#) do lado do servidor para seus dados paginação.

Por fim, AWS CLI tem filtragem do lado do [servidor e do lado do cliente que você pode usar individualmente ou em conjunto para filtrar](#) sua saída. AWS CLI

Tópicos

- [Saída sensível](#)
- [Opções de saída do lado do servidor e do lado do cliente](#)
- [Definição do formato da saída da AWS CLI](#)
- [Use as AWS CLI opções de paginação](#)
- [AWS CLI Saída do filtro](#)

Saída sensível

Algumas operações do AWS CLI podem retornar informações que podem ser consideradas confidenciais, incluindo informações de variáveis de ambiente. A exposição dessas informações pode representar um risco de segurança em determinados cenários; por exemplo, as informações podem ser incluídas nos registros de integração contínua e implantação contínua (CI/CD). Portanto, é importante que você revise quando está incluindo essa saída como parte de seus registros e suprima a saída quando não for necessária.

Para obter informações adicionais sobre a proteção de dados confidenciais, consulte [the section called “Proteção de dados”](#).

Considere as seguintes práticas recomendadas:

- Considere recuperar programaticamente seus segredos de um repositório de segredos, como. AWS Secrets Manager
- Revise o conteúdo dos seus registros de compilação para garantir que eles não contenham informações confidenciais. Considere abordagens como canalizar `/dev/null` ou capturar a saída como um bash ou PowerShell variável para suprimir as saídas do comando.

Veja a seguir um exemplo de bash para redirecionar a saída, mas não os erros, para: `/dev/null`

```
$ aws s3 ls > /dev/null
```

Para obter detalhes sobre como suprimir a saída do seu terminal, consulte a documentação do usuário do terminal que você usa.

- Considere o acesso aos seus registros e defina o escopo do acesso de acordo com seu caso de uso.

Opções de saída do lado do servidor e do lado do cliente

O AWS CLI tem filtragem do [lado do servidor e do lado do cliente que você pode usar individualmente ou em conjunto para filtrar](#) sua saída. A filtragem no lado do servidor é processada primeiro e retorna a saída para a filtragem no lado do cliente. A filtragem no lado do servidor é compatível com a API do serviço. A filtragem do lado do cliente é suportada pelo AWS CLI cliente usando o parâmetro. `--query`

As opções de saída do lado do servidor são recursos diretamente suportados pela API. AWS service (Serviço da AWS) Os dados filtrados ou paginados não são enviados ao cliente, o que pode acelerar os tempos de resposta HTTP e melhorar a largura de banda para conjuntos de dados maiores.

As opções de saída do lado do cliente são recursos criados pela AWS CLI. Todos os dados são enviados ao cliente e, em seguida, os AWS CLI filtros ou páginas são exibidos no conteúdo. As operações do lado do cliente não economizam velocidade ou largura de banda para conjuntos de dados maiores.

Quando as opções do lado do servidor e do lado do cliente são usadas juntas, as operações do lado do servidor são concluídas primeiro e depois enviadas ao cliente para operações do lado do cliente. Essa ação usa a economia em potencial de velocidade e largura de banda das opções do lado do servidor, enquanto usa recursos adicionais da AWS CLI para obter a saída desejada.

Definição do formato da saída da AWS CLI

Este tópico descreve os diferentes formatos de saída para a AWS Command Line Interface (AWS CLI). A AWS CLI oferece suporte aos seguintes formatos de saída:

- **json**: a saída é formatada como uma string [JSON](#).
- **text**: a saída é formatada como várias linhas de valores de string separados por tabulação. Isso pode ser útil para passar a saída para um processador de texto, como `grep`, `sed` ou `awk`.
- **table**: a saída é formatada como uma tabela usando os caracteres `+|-` para formar as bordas da célula. Geralmente, a informação é apresentada em um formato "amigável", que é muito mais fácil de ler do que outros, mas não tão útil programaticamente.

Como selecionar o formato de saída

Como explicado no tópico sobre [configuração](#), você pode especificar o formato de saída de três maneiras:

- Usando a opção **output** em um perfil nomeado no arquivo **config**: o exemplo a seguir ajusta o formato de saída padrão como `text`.

```
[default]
output=text
```

- Usando a variável de ambiente **AWS_DEFAULT_OUTPUT**: a saída a seguir ajusta o formato como `table` para os comandos nessa sessão de linhas de comandos até que a variável seja alterada ou a sessão encerrada. Usar essa variável de ambiente substitui qualquer valor definido no arquivo `config`.

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- Usando a opção **--output** na linha de comando: o exemplo a seguir define a saída apenas deste comando como `json`. Usar essa opção no comando substitui qualquer variável de ambiente definida atualmente ou o valor no arquivo `config`.

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

⚠ Important

O tipo de saída especificado por você modifica a forma de operação da opção `--query`.

- Se você especificar `--output text`, a saída será paginada antes que o filtro `--query` seja aplicado, e a AWS CLI executará a consulta uma vez em cada página da saída. Consequentemente, a consulta inclui o primeiro elemento correspondente em cada página, o que pode resultar em uma saída adicional inesperada. Para filtrar ainda mais a saída, é possível usar outras ferramentas da linha de comando, como `head` ou `tail`.
- Se você especificar `--output json`, a saída será completamente processada como uma única estrutura JSON nativa antes que o filtro `--query` seja aplicado. A AWS CLI executa a consulta apenas uma vez com relação a toda a estrutura, produzindo um resultado filtrado que é, então, a saída.

Formato de saída JSON

[JSON](#) é o formato de saída padrão da AWS CLI. A maioria das linguagens de programação pode facilmente decodificar strings JSON usando funções internas ou com bibliotecas disponíveis publicamente. É possível combinar a saída JSON com a [opção `--query`](#) de maneiras poderosas para filtrar e formatar a saída em formato JSON da AWS CLI.

Para filtragem mais avançada que talvez você não consiga fazer com `--query`, é possível considerar `jq`, um processador JSON da linha de comando. Baixe e localize o tutorial oficial em <http://stedolan.github.io/jq/>.

Veja a seguir um exemplo de saída JSON.

```
$ 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 de saída de texto

O formato `text` organiza a saída da AWS CLI em linhas delimitadas por tabulação. Isso funciona bem com ferramentas de texto Unix tradicionais, como `grep`, `sed` e `awk`, bem como o processamento de texto executado pelo PowerShell.

O formato de saída `text` segue a estrutura básica mostrada abaixo. As colunas são classificadas alfabeticamente por nomes de chaves correspondentes subjacentes do objeto JSON.

```

IDENTIFIER  sorted-column1 sorted-column2
IDENTIFIER2 sorted-column1 sorted-column2

```

A seguir, veja um exemplo de saída `text`. Cada campo é separado dos outros por tabulação, com uma guia adicional na qual há um campo vazio.

```
$ 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

```

A quarta coluna é o campo `PasswordLastUsed` e está vazia nas duas últimas entradas porque esses usuários nunca fazem login no AWS Management Console.

⚠ Important

É altamente recomendável que, se você especificar uma saída `text`, também use sempre a opção `--query` para garantir um comportamento consistente.

Isso ocorre porque o formato de texto ordena alfabeticamente as colunas da saída pelo nome da chave do objeto JSON subjacente retornado pelo serviço da AWS, e recursos similares podem não têm os mesmos nomes de chave. Por exemplo, a representação JSON de uma instância do Amazon EC2 baseada em Linux pode ter elementos que não estão presentes na representação JSON de uma instância baseada em Windows ou vice-versa. Além disso, os recursos podem ter elementos de valores de chave adicionados ou removidos em futuras atualizações, alterando a ordem das colunas. Este é o local em que `--query` expande a funcionalidade da saída `text` para fornecer total controle sobre o formato de saída.

No exemplo a seguir, o comando especifica quais elementos devem ser exibidos e define a ordem das colunas com a notação da lista `[key1, key2, ...]`. Isso oferece a você a segurança de que os valores de chave corretos sempre serão exibido na coluna esperada. Por fim, observe como a AWS CLI resulta em `None` como os valores para as chaves que não existem.

```
$ 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

```


O exemplo a seguir mostra como você pode usar `grep` e `awk` com a saída `text` do comando `aws ec2 describe-instances`. O primeiro comando exibe a zona de disponibilidade, o estado atual e o ID de cada instância na saída `text`. O segundo comando processa essa saída para exibir somente os IDs de todas as instâncias em execução na Zona de disponibilidade `us-west-2a`.

```
$ 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
```

O exemplo a seguir vai mais longe e mostra não apenas como filtrar a saída, mas como usar essa saída para automatizar a alteração dos tipos de instância para cada instância interrompida.

```
$ 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
```

A saída `text` também pode ser útil no PowerShell. Como as colunas na saída `text` são delimitadas por tabulação, é possível dividir facilmente a saída em uma matriz usando o delimitador ``t` do PowerShell. O comando a seguir exibe o valor da terceira coluna (`InstanceId`) se a primeira coluna (`AvailabilityZone`) corresponder à string `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
```

Embora o exemplo anterior mostre como usar o parâmetro `--query` para analisar os objetos JSON subjacentes e extrair a coluna desejada, o PowerShell tem capacidade própria para lidar com o JSON, se a compatibilidade entre as plataformas não for uma preocupação. Em vez de lidar com a saída como texto, o que é exigido pela maioria dos shells de comando, o PowerShell permite que você use o cmdlet `ConvertFrom-JSON` para produzir um objeto estruturado hierarquicamente. Depois, é possível acessar diretamente o membro desejado usando esse objeto.

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

Tip

Se você enviar o texto e filtrar a saída para um único campo usando o parâmetro `--query`, a saída será uma única linha de valores separados por tabulação. Para obter cada valor em uma linha separada, você pode colocar o campo de saída entre colchetes, conforme mostrado nos exemplos a seguir.

Saída de linha única separada por tabulação:

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].GroupName"
```

```
HRDepartment    Developers      SpreadsheetUsers  LocalAdmins
```

Cada valor em sua própria linha, colocando `[GroupName]` entre colchetes:

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].[GroupName]"
```

```

HRDepartment
Developers
SpreadsheetUsers
LocalAdmins

```

Formato de saída de tabela

O formato `table` produz representações legíveis da saída complexa da AWS CLI em um formato tabular.

```
$ 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 ||
+-----+-----+-----+-----+|
+

```

É possível combinar a opção `--query` com o formato `table` para exibir um conjunto de elementos pré-selecionado na saída bruta. As diferenças da saída entre as notações de lista e dicionário: no primeiro exemplo, os nomes de coluna são ordenados alfabeticamente, e, no segundo, as colunas

sem nome são ordenadas conforme definido pelo usuário. Para obter mais informações sobre a opção `--query`, consulte [AWS CLI Saída do filtro](#).

```
$ 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  |
+-----+-----+-----+-----+
```

Use as AWS CLI opções de paginação

Este tópico descreve as diferentes maneiras de paginar a saída pela AWS CLI.

Paginação do lado do servidor

Para comandos que podem retornar uma grande lista de itens, o AWS Command Line Interface (AWS CLI) tem várias opções para controlar o número de itens incluídos na saída quando AWS CLI chama um serviço API para preencher a lista.

As opções incluem o seguinte:

- [Como usar o parâmetro `--no-paginate`](#)
- [Como usar o parâmetro `--page-size`](#)

- [Como usar o parâmetro `--max-items`](#)
- [Como usar o parâmetro `--starting-token`](#)

Por padrão, o AWS CLI usa um tamanho de página determinado pelo serviço individual e recupera todos os itens disponíveis. Por exemplo, o Amazon S3 tem um tamanho de página padrão de mil. Se você executar `aws s3api list-objects` em um bucket do Amazon S3 que contém 3,5 mil objetos, a AWS CLI fará automaticamente quatro chamadas para o Amazon S3, lidando com a lógica de paginação específica do serviço para você em segundo plano e retornando todos os 3,5 mil objetos na saída final.

Como usar o parâmetro `--no-paginate`

A opção `--no-paginate` desabilita os seguintes tokens de paginação no lado do cliente. Ao usar um comando, por padrão, ele faz AWS CLI automaticamente várias chamadas para retornar todos os resultados possíveis para criar paginação. Uma chamada para cada página. A desativação da paginação tem a AWS CLI única chamada uma vez para a primeira página dos resultados do comando.

Por exemplo, se você executa `aws s3api list-objects` em um bucket do Amazon S3 que contém 3.500 objetos, ele AWS CLI só faz a primeira chamada para o Amazon S3, retornando somente os primeiros 1.000 objetos na saída final.

```
$ aws s3api list-objects \  
  --bucket my-bucket \  
  --no-paginate  
{  
  "Contents": [  
  ...
```

Como usar o parâmetro `--page-size`

Caso haja problemas ao executar os comandos da lista em um grande número de recursos, o tamanho da página padrão poderá ser muito alto. Isso pode fazer com que as chamadas para AWS serviços excedam o tempo máximo permitido e gerem um erro de “tempo limite”. Você pode usar a `--page-size` opção para especificar que AWS CLI solicite um número menor de itens de cada chamada para o AWS serviço. O AWS CLI ainda recupera a lista completa, mas executa um número maior de API chamadas de serviço em segundo plano e recupera um número menor de itens com cada chamada. Isso fornece às chamadas individuais chances melhores de sucesso sem um tempo

limite. Alterar o tamanho da página não afeta a saída; afeta somente o número de API chamadas que precisam ser feitas para gerar a saída.

```
$ aws s3api list-objects \  
  --bucket my-bucket \  
  --page-size 100  
{  
  "Contents": [  
  ...
```

Como usar o parâmetro --max-items

Para incluir menos itens por vez na AWS CLI saída, use a `--max-items` opção. O AWS CLI Still processa a paginação com o serviço conforme descrito anteriormente, mas imprime somente o número de itens por vez que você especifica.

```
$ aws s3api list-objects \  
  --bucket my-bucket \  
  --max-items 100  
{  
  "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==",  
  "Contents": [  
  ...
```

Como usar o parâmetro --starting-token

Se o número de itens output (`--max-items`) for menor que o número total de itens retornados pelas API chamadas subjacentes, a saída incluirá um `NextToken` que você pode passar para um comando subsequente para recuperar o próximo conjunto de itens. O exemplo a seguir mostra como usar o valor `NextToken` retornado pelo exemplo anterior e permite que você recupere os segundos 100 itens.

Note

O parâmetro `--starting-token` não pode ser nulo nem vazio. Se o comando anterior não retornar um valor `NextToken`, não haverá mais itens a serem retornados e você não precisará chamar o comando novamente.

```
$ aws s3api list-objects \  
  --starting-token
```

```
--bucket my-bucket \  
--max-items 100 \  
--starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==  
{  
  "Contents": [  
    ...
```

O AWS serviço especificado pode não devolver itens no mesmo pedido sempre que você ligar. Se você especificar valores diferentes para `--page-size` e `--max-items`, poderá obter resultados inesperados com itens ausentes ou duplicados. Para evitar que isso aconteça, use o mesmo número para `--page-size` e `--max-items` para sincronizar a paginação da AWS CLI com a paginação do serviço subjacente. Também é possível recuperar a lista completa e executar quaisquer operações necessárias de paginação localmente.

AWS CLI Saída do filtro

O AWS Command Line Interface (AWS CLI) tem filtragem do lado do servidor e do lado do cliente que você pode usar individualmente ou em conjunto para filtrar sua saída. A filtragem no lado do servidor é processada primeiro e retorna a saída para a filtragem no lado do cliente.

- A filtragem do lado do servidor é suportada pelo API, e você geralmente a implementa com um parâmetro. `--filter` O serviço retorna apenas resultados correspondentes, o que pode acelerar os tempos de HTTP resposta para grandes conjuntos de dados.
- A filtragem do lado do cliente é suportada pelo AWS CLI cliente usando o parâmetro. `--query` Esse parâmetro oferece recursos que a filtragem do lado do servidor pode não ter.

Tópicos

- [Filtragem no lado do servidor](#)
- [Filtragem no lado do cliente](#)
- [Combinação das filtragens no lado do servidor e no lado do cliente](#)
- [Recursos adicionais](#)

Filtragem no lado do servidor

A filtragem do lado do servidor no AWS CLI é fornecida pelo serviço. AWS API O AWS serviço retorna apenas os registros na HTTP resposta que correspondem ao seu filtro, o que pode acelerar os tempos de HTTP resposta para grandes conjuntos de dados. Como a filtragem do lado do

servidor é definida pelo serviço API, os nomes e funções dos parâmetros variam entre os serviços. Alguns nomes de parâmetros comuns usados na filtragem são:

- `--filter` como [ses](#) e [ce](#).
- `--filters` como [ec2](#), [autoescalabilidade](#) e [rds](#).
- Nomes que começam com a palavra `filter`, por exemplo `--filter-expression`, para o comando [aws dynamodb scan](#).

Filtragem no lado do cliente

AWS CLI Ele fornece recursos integrados JSON de filtragem do lado do cliente com o parâmetro.

`--query` O parâmetro `--query` é uma ferramenta poderosa que você pode usar para personalizar o conteúdo e estilo da sua saída. O `--query` parâmetro pega a HTTP resposta que retorna do servidor e filtra os resultados antes de exibi-los. Como toda a HTTP resposta é enviada ao cliente antes da filtragem, a filtragem do lado do cliente pode ser mais lenta do que a filtragem do lado do servidor para grandes conjuntos de dados.

A consulta usa a [JMESPathsyntaxe](#) para criar expressões para filtrar sua saída. Para aprender a JMESPath sintaxe, consulte o [Tutorial](#) no JMESPathsite.

Important

O tipo de saída especificado por você modifica a forma de operação da opção `--query`.

- Se você especificar `--output text`, a saída será paginada antes da aplicação do `--query` filtro e AWS CLI executará a consulta uma vez em cada página da saída. Consequentemente, a consulta inclui o primeiro elemento correspondente em cada página, o que pode resultar em uma saída adicional inesperada. Para filtrar ainda mais a saída, é possível usar outras ferramentas da linha de comando, como `head` ou `tail`.
- Se você especificar `--output json`, a saída será completamente processada como uma única estrutura JSON nativa antes que o filtro `--query` seja aplicado. Ele AWS CLI executa a consulta apenas uma vez em toda a estrutura, produzindo um resultado filtrado que é então gerado.

Tópicos de filtragem no lado do cliente

- [Antes de começar](#)

- [Identificadores](#)
- [Como selecionar em uma lista](#)
- [Como filtrar dados aninhados](#)
- [Resultados da simplificação](#)
- [Filtragem de valores específicos](#)
- [Encadeamento de expressões](#)
- [Filtragem por vários valores de identificador](#)
- [Adição de rótulos a valores de identificador](#)
- [Funções](#)
- [Exemplos avançados --query](#)

Antes de começar

Ao usar expressões de filtro usadas nesses exemplos, certifique-se de usar as regras de aspas corretas para o shell do terminal. Para ter mais informações, consulte [the section called “Aspas com strings”](#).

A JSON saída a seguir mostra um exemplo do que o `--query` parâmetro pode produzir. O resultado descreve três EBS volumes da Amazon anexados a EC2 instâncias separadas da Amazon.

Exemplo de saída

```
$ 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
}
]
```

```
}
```

Identificadores

Identificadores são rótulos para valores de saída. Ao criar filtros, você usa identificadores para refinar os resultados da consulta. No exemplo de saída a seguir, todos os identificadores como `Volumes`, `AvailabilityZone` e `AttachTime` estão destacados.

```
$ 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
  }
]
}

```

Para obter mais informações, consulte [Identificadores](#) no JMESPathsite.

Como selecionar em uma lista

Uma lista ou matriz é um identificador que é seguido por um colchete “[”, como `Volumes` e `Attachments` na [the section called “Antes de começar”](#).

Sintaxe

```
<listName>[ ]
```

Para filtrar toda a saída de uma matriz, você pode usar a notação curinga. Expressões [curinga](#) são expressões usadas para retornar elementos usando a notação `*`.

O exemplo a seguir consulta todo o conteúdo `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
  }
]
```

Para exibir um volume específico na matriz por índice, chame o índice da matriz. Por exemplo, o primeiro item na matriz `Volumes` tem um índice igual a 0, o que resulta na consulta `Volumes[0]`. Para obter mais informações sobre índices de matriz, consulte [expressões de índice](#) no JMESPathsite.

```
$ 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
}
```

Para exibir um intervalo específico de volumes por índice, use `slice` com a seguinte sintaxe, onde `start` é o índice da matriz inicial, `stop` é o índice onde o filtro interrompe o processamento e `step` é o intervalo de salto.

Sintaxe

```
<arrayName>[<start>:<stop>:<step>]
```

Se algum deles for omitido da expressão `slice`, os seguintes valores padrão serão usados:

- Start: o primeiro índice na lista, 0.
- Stop: o último índice na lista.
- Step: sem pular etapas, onde o valor é 1.

Para retornar somente os dois primeiros volumes, use um valor de start igual a 0, um valor de stop igual 2 e um valor de step igual a 1, conforme mostrado no exemplo a seguir.

```
$ 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
  }
]
```

```
}
]
```

Como este exemplo contém valores padrão, você pode encurtar o slice de `Volumes[0:2:1]` para `Volumes[:2]`.

O exemplo a seguir omite valores padrão e retorna cada dois volumes em toda a matriz.

```
$ 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
}
]
```

Os valores de `step` também podem ser números negativos, o que resulta na filtragem em ordem inversa de uma matriz, conforme mostrado no exemplo a seguir.

```
$ 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
}
]

```

Para obter mais informações, consulte [Slices](#) no JMESPathsite.

Como filtrar dados aninhados

Para refinar a filtragem de `Volumes[*]` para valores aninhados, use subexpressões anexando um ponto e seus critérios de filtragem.

Sintaxe

```
<expression>.<expression>
```

O exemplo a seguir mostra todas as informações de `Attachments` para todos os volumes.

```

$ 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"
  }
]
]
```

Para filtrar ainda mais os valores aninhados, acrescente a expressão para cada identificador aninhado. O exemplo a seguir lista o State para todos os Volumes.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

Resultados da simplificação

Para obter mais informações, consulte [SubExpressions](#) no JMESPathsite.

Você pode simplificar os resultados para `Volumes[*].Attachments[*].State` removendo a notação curinga, o que resultará na consulta `Volumes[*].Attachments[].State`. A simplificação muitas vezes é útil para melhorar a legibilidade dos resultados.

```

$ aws ec2 describe-volumes \
```

```

--query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]

```

Para obter mais informações, consulte [Flatten](#) no JMESPathsite.

Filtragem de valores específicos

Para filtrar valores específicos em uma lista, use uma expressão de filtro como mostrado na sintaxe a seguir.

Sintaxe

```
? <expression> <comparator> <expression>]
```

Os comparadores de expressão incluem ==, !=, <, <=, > e >=. O exemplo a seguir filtra VolumeIds para todos os Volumes em um State igual a Attached.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'
[
  [
    "vol-e11a5288"
  ],
  [
    "vol-2e410a47"
  ],
  [
    "vol-a1b3c7nd"
  ]
]

```

Os resultados podem ser simplificados, o que resulta no exemplo a seguir.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'
[
  "vol-e11a5288",

```

```
"vol-2e410a47",  
"vol-a1b3c7nd"  
]
```

O exemplo a seguir filtra VolumeIds para todos os Volumes com tamanho inferior a 20.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[?Size < `20`].VolumeId'  
[  
  "vol-2e410a47",  
  "vol-a1b3c7nd"  
]
```

Para obter mais informações, consulte [Filtrar expressões](#) no JMESPathsite.

Encadeamento de expressões

Você pode encadear resultados de um filtro para uma nova lista e, em seguida, filtrar o resultado com outra expressão usando a seguinte sintaxe:

Sintaxe

```
<expression> | <expression>]
```

O exemplo a seguir leva os resultados do filtro da expressão `Volumes[*].Attachments[].InstanceId` e produz o primeiro resultado na matriz.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId | [0]'  
"i-a071c394"
```

Esse exemplo faz isso criando primeiro a matriz a partir da expressão a seguir.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId'  
"i-a071c394",  
"i-4b41a37c",  
"i-1jd73kv8"
```

Depois, devolva o primeiro elemento para essa matriz.

```
"i-a071c394"
```

Para obter mais informações, consulte [Pipe Expressions](#) no JMESPathsite.

Filtragem por vários valores de identificador

Para filtrar por vários identificadores, use uma lista de seleção múltipla usando a seguinte sintaxe:

Sintaxe

```
<listName>[].[<expression>, <expression>]
```

No exemplo a seguir, `VolumeId` e `VolumeType` são filtrados na lista `Volumes`, o que resulta na expressão abaixo.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[].[VolumeId, VolumeType]'
[
  [
    "vol-e11a5288",
    "standard"
  ],
  [
    "vol-2e410a47",
    "standard"
  ],
  [
    "vol-a1b3c7nd",
    "standard"
  ]
]
```

Para adicionar dados aninhados à lista, adicione outra lista de seleção múltipla. O exemplo a seguir expande o exemplo anterior filtrando também por `InstanceId` e `State` na lista aninhada `Attachments`. Isso resulta na seguinte expressão.

```
$ 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"
      ]
    ]
  ]
]

```

Para torná-la mais legível, simplifique a expressão conforme mostrado no exemplo a seguir.

```

$ 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"
]
]
```

Para obter mais informações, consulte [Lista de seleção múltipla](#) no JMESPathsite.

Adição de rótulos a valores de identificador

Para tornar essa saída mais fácil de ler, use um hash de seleção múltipla com a sintaxe a seguir.

Sintaxe

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

O rótulo do identificador não precisa ser o mesmo que o nome do identificador. O exemplo a seguir usa o rótulo `VolumeType` para os valores `VolumeType`.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[].{VolumeType: VolumeType}'
[
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  }
]
```

Para simplificar, o exemplo a seguir mantém os nomes dos identificadores para cada rótulo e exibe `VolumeId`, `VolumeType`, `InstanceId` e `State` para todos os volumes:


```
$ 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"
  }
]
```

Para obter mais informações, consulte [Hash de seleção múltipla JMESPath](#) no site.

Funções

A JMESPath sintaxe contém muitas funções que você pode usar para suas consultas. Para obter informações sobre JMESPath funções, consulte [Funções integradas](#) no JMESPathsite.

Para demonstrar como você pode incorporar uma função em suas consultas, o exemplo a seguir usa a função `sort_by`. A função `sort_by` classifica uma matriz usando uma expressão como a chave de classificação com a seguinte sintaxe:

Sintaxe

```
sort_by(<listName>, <sort expression>)[].<expression>
```

O exemplo a seguir usa o [exemplo de hash de multisseleção](#) anterior e classifica a saída por `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"
  }
]
```

Para obter mais informações, consulte [sort_by no site](#). JMESPath

Exemplos avançados --query

Para extrair informações de um item específico

O exemplo a seguir usa o parâmetro --query para localizar um item específico em uma lista e extrai as informações desse item. O exemplo lista todas as AvailabilityZones associadas ao endpoint de serviço especificado. Ele extrai o item da lista ServiceDetails que tem o ServiceName especificado e gera o campo AvailabilityZones do item selecionado.

```
$ 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"
  ]
]
```

Para mostrar snapshots após a data de criação especificada

O exemplo a seguir mostra como listar todos os seus snapshots que foram criados após uma data especificada, incluindo apenas alguns dos campos disponíveis na saída.

```
$ 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
  }
]
```

Para mostrar o mais recente AMIs

O exemplo a seguir lista as cinco Amazon Machine Images (AMIs) mais recentes que você criou, classificadas da mais recente para a mais antiga.

```
$ 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"
  }
]
```

Para mostrar instâncias não íntegras do Auto Scaling

O exemplo a seguir mostra apenas o InstanceId para todas as instâncias com problemas de integridade no grupo de AutoScaling especificado.

```
$ aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-name My-AutoScaling-Group-Name \
  --output text \
  --query 'AutoScalingGroups[*].Instances[?HealthStatus==`Unhealthy`].InstanceId'
```

Para incluir volumes com a tag especificada

O exemplo a seguir descreve todas as instâncias com uma tag test. Contudo que haja outra etiqueta além de test anexada ao volume, o volume ainda é retornado nos resultados.

A expressão abaixo para retornar todas as tags com a tag test em uma matriz. Qualquer etiqueta que não é a etiqueta test contém um valor null.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Tags[?Value == `test`]'
```

Para excluir volumes com a etiqueta especificada

O exemplo a seguir descreve todas as instâncias sem uma etiqueta test. A utilização de uma expressão ?Value != `test` simples não funciona para excluir um volume, pois os volumes podem ter várias etiquetas. Contudo que haja outra etiqueta além de test anexada ao volume, o volume ainda é retornado nos resultados.

Para excluir todos os volumes com a etiqueta `test`, comece com a expressão abaixo para retornar todas as etiquetas com a etiqueta `test` em um array. Qualquer etiqueta que não é a etiqueta `test` contém um valor `null`.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

Em seguida, filtrar todos os resultados `test` positivos usando a função `not_null`.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)]'
```

Encadeie os resultados para simplificá-los, o que resultará na consulta a seguir.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)] | []'
```

Combinação das filtragens no lado do servidor e no lado do cliente

É possível usar as filtragens no lado do servidor e no lado do cliente ao mesmo tempo. A filtragem no lado do servidor é concluída primeiro, o que envia os dados para o cliente que, em seguida, são filtrados pelo parâmetro `--query`. Se você estiver usando grandes conjuntos de dados, usar primeiro a filtragem do lado do servidor pode reduzir a quantidade de dados enviados ao cliente para cada AWS CLI chamada, mantendo a personalização poderosa que a filtragem do lado do cliente oferece.

O exemplo a seguir lista os EC2 volumes da Amazon usando filtragem do lado do servidor e do lado do cliente. O serviço filtra uma lista de todos os volumes anexados na zona de disponibilidade `us-west-2a`. O parâmetro `--query` limita ainda mais a saída apenas para aqueles volumes com um valor `Size` maior que 50 e mostra apenas os campos especificados com nomes definidos pelo usuário.

```
$ 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"  
    }  
]
```

O exemplo a seguir recupera uma lista de imagens que atendem a vários critérios. Depois, usa o parâmetro `--query` para classificar a saída por `CreationDate`, selecionando apenas as mais recentes. Por fim, ele exibe o `ImageId` dessa imagem.

```
$ aws ec2 describe-images \  
  --owners amazon \  
  --filters "Name=name,Values=amzn*gp2" "Name=virtualization-type,Values=hvm"  
  "Name=root-device-type,Values=ebs" \  
  --query "sort_by(Images, &CreationDate)[-1].ImageId" \  
  --output text  
ami-00ced3122871a4921
```

O exemplo a seguir exibe o número de volumes disponíveis que são mais de 1000 IOPS usando `length` para contar quantos estão em uma lista.

```
$ aws ec2 describe-volumes \  
  --filters "Name=status,Values=available" \  
  --query 'length(Volumes[?Iops > `1000`])'  
3
```

Recursos adicionais

JMESPathTerminal

JMESPathTerminal é um comando de terminal interativo para experimentar JMESPath expressões usadas para filtragem do lado do cliente. Usando o comando `jpterm`, o terminal mostra os resultados imediatos da consulta enquanto você está digitando. Você pode canalizar a AWS CLI saída diretamente para o terminal, permitindo a experimentação avançada de consultas.

O exemplo a seguir canaliza a `aws ec2 describe-volumes` saída diretamente para o JMESPath Terminal.

```
$ aws ec2 describe-volumes | jpterm
```

Para obter mais informações sobre o JMESPath Terminal e as instruções de instalação, consulte [JMESPathTerminal](#) ativado GitHub.

Utilitário jq

O jq fornece uma maneira de transformar sua saída no lado do cliente em um formato de saída desejado. Para obter mais informações jq e instruções de instalação, consulte [jq](#) on GitHub.

Códigos de retorno da AWS CLI

O código de retorno geralmente é um código oculto enviado após a execução de um comando da AWS Command Line Interface(AWS CLI) que descreve o status do comando. Você pode usar o comando `echo` para exibir o código enviado do último comando AWS CLI e usar esses códigos para determinar se um comando foi bem-sucedido ou se falhou, e por que um comando pode ter apresentado um erro. Além dos códigos de retorno, você pode visualizar mais detalhes sobre uma falha executando seus comandos com o a opção `--debug`. Essa opção produz um relatório detalhado das etapas que a AWS CLI usa para processar o comando, e o resultado que foi gerado por cada etapa.

Para determinar o código de retorno de um comando de AWS CLI, execute um dos seguintes comandos imediatamente após a execução do comando de CLI.

Linux and macOS

```
$ echo $?  
0
```

Windows PowerShell

```
PS> echo $lastexitcode  
0
```

Windows Command Prompt

```
C:\> echo %errorlevel%  
0
```

Veja a seguir os valores de código de retorno que podem ser retornados ao final da execução de um comando de AWS Command Line Interface (AWS CLI).

Código	Significado
0	O serviço respondeu com um código de status de resposta HTTP de 200, o que indica que não houve erros gerados pela AWS CLI e pelo serviço da AWS para o qual a solicitação foi enviada.
1	Uma ou mais operações de transferência do Amazon S3 falhou. Limitado a comandos do S3.
2	O significado desse código de retorno depende do comando. <ul style="list-style-type: none">• Aplicável a todos os comandos da AWS CLI : não foi possível analisar o comando inserido. Falhas de análise podem ser causadas, entre outros motivos, pela ausência de subcomandos ou argumentos necessários ou pelo uso de comandos ou argumentos desconhecidos.• Limitado a comandos do S3: um ou mais arquivos marcados para transferência foram ignorados durante o processo de transferência. No entanto, todos os outros arquivos marcados para transferência foram transferidos com êxito. Os arquivos que são ignorados durante o processo de transferência incluem: arquivos inexistentes, arquivos que são dispositivos de caracteres especiais, dispositivo de bloqueio especial, filas FIFO ou soquetes, além de arquivos para os quais o usuário não tem permissões de leitura.
130	O comando foi interrompido por um SIGINT. Este é o sinal enviado por você para cancelar um comando com <code>Ctrl+C</code> .
255	Ocorreu uma falha no comando. Houve erros gerados pela AWS CLI ou pelo serviço da AWS para o qual a solicitação foi enviada.

Crie e use atalhos de AWS CLI comando chamados aliases

Os aliases são atalhos que você pode criar no AWS Command Line Interface (AWS CLI) para encurtar comandos ou scripts que você usa com frequência. Os aliases são criados no arquivo `alias` localizado em sua pasta de configuração.

Tópicos

- [Pré-requisitos](#)
- [Etapa 1: Criação do arquivo de alias](#)

- [Etapa 2: Criação de um alias](#)
- [Passo 3: Como chamar um alias](#)
- [Exemplos de repositório de alias](#)
- [Recursos](#)

Pré-requisitos

Para usar comandos de alias, é necessário fazer o seguinte:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI e Autenticação e credenciais de acesso](#).
- Use uma AWS CLI versão mínima de 1.11.24 ou 2.0.0.
- (Opcional) Para usar scripts bash de AWS CLI alias, você deve usar um terminal compatível com bash.

Etapa 1: Criação do arquivo de alias

Para criar o alias arquivo, você pode usar sua navegação de arquivos e um editor de texto, ou usar seu terminal preferido usando o step-by-step procedimento. Para criar rapidamente seu arquivo de alias, use o seguinte bloco de comandos.

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
```

Para criar o arquivo de alias

1. Crie uma pasta chamada `cli` na sua pasta AWS CLI de configuração. Por padrão, a pasta de configuração é `~/.aws/` no Linux ou no macOS e `%USERPROFILE%\aws\` no Windows. Você pode criar isso via sua navegação de arquivos ou usando o comando a seguir.

Linux and macOS

```
$ mkdir -p ~/.aws/cli
```

Windows

```
C:\> md %USERPROFILE%\.aws\cli
```

O caminho padrão da pasta `cli` resultante é `~/.aws/cli/` no Linux ou no macOS e `%USERPROFILE%\.aws\cli` no Windows.

2. Na pasta `cli`, crie um arquivo de texto chamado `aliassem` extensão e adicione `[toplevel]` à primeira linha. Você pode criar esse arquivo com seu editor de texto preferido ou usar o comando a seguir.

Linux and macOS

```
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> echo [toplevel] > %USERPROFILE%\.aws\cli\alias
```

Etapa 2: Criação de um alias

Você pode criar um alias usando comandos básicos ou scripts bash.

Criação um alias de comando básico

Você pode criar seu alias ao adicionar um comando usando a seguinte sintaxe no arquivo `alias` criado na etapa anterior.

Sintaxe

```
aliasname = command [--options]
```

A ferramenta *aliasname* é o que você chama de seu pseudônimo. A ferramenta *command* é o comando que você deseja chamar, que pode incluir outros aliases. Você pode incluir opções ou parâmetros em seu alias ou adicioná-los ao chamar seu alias.

O exemplo a seguir cria um alias chamado `aws whoami` usando o comando [aws sts get-caller-identity](#). Como esse alias chama um comando da AWS CLI existente, você pode escrever o comando sem usar o prefixo `aws`.

```
whoami = sts get-caller-identity
```

O exemplo a seguir usa o exemplo `whoami` anterior e adiciona o filtro `Account` e as opções `output` de texto.

```
whoami2 = sts get-caller-identity --query Account --output text
```

Criar um alias de subcomando

Note

O recurso de alias do subcomando requer uma AWS CLI versão mínima de 1.11.24 ou 2.0.0

É possível criar um alias para subcomandos adicionando um comando usando a seguinte sintaxe no arquivo `alias` criado na etapa anterior.

Sintaxe

```
[command commandGroup]  
aliasname = command [--options]
```

A ferramenta *commandGroup* é o namespace do comando, por exemplo, o comando `aws ec2 describe-regions` está no grupo de `ec2` comandos. A ferramenta *aliasname* é o que você chama de seu pseudônimo. A ferramenta *command* é o comando que você deseja chamar, que pode incluir outros aliases. Você pode incluir opções ou parâmetros em seu alias ou adicioná-los ao chamar seu alias.

O exemplo a seguir cria um alias chamado `aws ec2 regions` usando o comando [aws ec2 describe-regions](#). Como esse alias chama um comando da AWS CLI existente sob o namespace do comando `ec2`, é possível escrever o comando sem usar o prefixo `aws ec2`.

```
[command ec2]
regions = describe-regions --query Regions[].RegionName
```

Para criar aliases de comandos fora do namespace do comando, use um ponto de exclamação como prefixo do comando completo. O exemplo a seguir cria um alias chamado `aws ec2 instance-profiles` usando o comando [aws iam list-instance-profiles](#).

```
[command ec2]
instance-profiles = !aws iam list-instance-profiles
```

Note

Os aliases usam apenas namespaces de comandos existentes e não é possível criar outros. Por exemplo, você não pode criar um alias com a seção `[command johnsmith]`, pois o namespace do comando `johnsmith` ainda não existe.

Criação de um alias de script bash

Warning

Para usar scripts bash de AWS CLI alias, você deve usar um terminal compatível com bash

Você pode criar um alias usando scripts bash para processos mais avançados usando a sintaxe a seguir.

Sintaxe

```
aliasname =
  !f() {
    script content
  }; f
```

A ferramenta *aliasname* é o que você chama de seu alias e *script content* é o script que você deseja executar ao chamar o alias.

O exemplo a seguir usa `opendns` para mostrar seu endereço IP atual. Como você pode usar aliases em outros aliases, o alias `myip` a seguir é útil para permitir ou revogar o acesso ao seu endereço IP de dentro de outros aliases.

```
myip =
  !f() {
    dig +short myip.opendns.com @resolver1.opendns.com
  }; f
```

O exemplo de script a seguir chama o `aws myip` alias anterior para autorizar seu endereço IP para a entrada de um grupo de EC2 segurança da 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 você chama aliases que usam scripts bash, as variáveis são sempre passadas na ordem em que você as inseriu. Nos scripts bash, os nomes das variáveis não são levados em consideração, apenas a ordem em que aparecem. No exemplo de alias `textalert` a seguir, a variável para a opção `--message` é a primeira e para a opção `--phone-number` é a segunda.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

Passo 3: Como chamar um alias

Para executar o alias que você criou em seu arquivo `alias`, use a sintaxe a seguir. Você pode adicionar opções extras ao chamar seu alias.

Sintaxe

```
$ aws aliasname
```

O exemplo a seguir usa o alias do comando `aws whoami`.

```
$ aws
  whoami
{
  "UserId": "A12BCD34E5FGHI6JKLM",
  "Account": "1234567890987",
  "Arn": "arn:aws:iam::1234567890987:user/userName"
}
```

O exemplo a seguir usa o alias `aws whoami` com opções adicionais para retornar somente o número `Account` na saída de `text`.

```
$ aws whoami --query Account --output
text
1234567890987
```

O exemplo a seguir usa o [alias do subcomando](#) `aws ec2 regions`.

```
$ aws ec2
  regions
[
  "ap-south-1",
  "eu-north-1",
  "eu-west-3",
  "eu-west-2",
  ...
]
```

Como chamar um alias usando variáveis de script bash

Quando você chama aliases que usam scripts bash, as variáveis são passadas na ordem em que foram inseridas. Nos scripts bash, os nomes das variáveis não são levados em consideração, apenas a ordem em que aparecem. Por exemplo, no alias `textalert` a seguir, a variável para a opção `--message` é a primeira e para a opção `--phone-number` é a segunda.

```
textalert =
!f() {
  aws sns publish --message "${1}" --phone-number ${2}
}; f
```

Ao chamar o alias `textalert`, você precisa passar variáveis na mesma ordem em que elas são executadas no alias. No exemplo a seguir usamos as variáveis `$message` e `$phone`. A variável

\$message é passada como \${1} para a opção --message e a variável \$phone é passada como \${2} para a opção --phone-number. Isso faz com que o alias `textalert` seja chamado com êxito para enviar uma mensagem.

```
$ aws textalert $message
  $phone
{
  "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

No exemplo a seguir, a ordem é invertida quando o alias é chamado para \$phone e \$message. A variável \$phone é passada como \${1} para a opção --message e a variável \$message é passada como \${2} para a opção --phone-number. Como as variáveis estão fora de ordem, o alias passa as variáveis incorretamente. Isso causa um erro porque o conteúdo de \$message não corresponde aos requisitos de formatação de número de telefone para a opção --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
```

Exemplos de repositório de alias

O [repositório AWS CLI de aliases](#) em GitHub contém exemplos de AWS CLI aliases criados pela equipe de AWS CLI desenvolvedores e pela comunidade. Você pode usar todo o arquivo de exemplo de `alias` ou use aliases individuais para seu próprio uso.

Warning

A execução dos comandos nesta seção exclui seu arquivo `alias`. Para evitar que o arquivo de alias existente seja sobrescrito, altere o local de download.

Para usar aliases do repositório

1. Instale o Git. Para obter instruções de instalação, consulte [Introdução: Instalação do Git](#) na Documentação do Git.
2. Instale o comando `jp`. O comando `jp` é usado no alias `tostring`. Para obter instruções de instalação, consulte [JMESPath\(jp\) README .md](#) em GitHub.
3. Instale o comando `jq`. O comando `jq` é usado no alias `tostring-with-jq`. Para obter instruções de instalação, consulte o [JSONprocessador \(jq\) ligado GitHub](#).
4. Baixe o `alias` de uma das seguintes formas:
 - Execute os seguintes comandos que são baixados do repositório e copiam o arquivo `alias` para sua pasta de configuração.

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
```

- Faça o download diretamente do repositório e salve na `cli` pasta da sua pasta AWS CLI de configuração. Por padrão, a pasta de configuração é `~/.aws/` no Linux ou no macOS e `%USERPROFILE%\aws\` no Windows.
5. Para verificar se os aliases estão funcionando, execute o `alias` a seguir.

```
$ aws whoami
```

Isso exibe a mesma resposta que o comando `aws sts get-caller-identity`:

```
{
  "Account": "012345678901",
  "UserId": "AIUAINBADX2VEG2TC6HD6",
  "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```


Recursos

- O [repositório AWS CLI de aliases](#) em GitHub contém exemplos de AWS CLI aliases criados pela equipe de AWS CLI desenvolvedores e a contribuição da comunidade. AWS CLI
- O anúncio do recurso de alias de [AWS re:Invent 2016: The Effective User on. AWS CLI YouTube](#)
- [aws sts get-caller-identity](#)
- [aws ec2 describe-instances](#)
- [aws sns publish](#)

Exemplos de código

Este capítulo fornece uma coleção de exemplos que mostram como usar o AWS Command Line Interface (AWS CLI) com Serviços da AWS.

O AWS CLI tem os seguintes tipos de exemplos neste guia:

- [Exemplos de comandos guiados](#) - Exemplos de comandos guiados para o Guia AWS CLI do Usuário sobre como usar o AWS CLI com alguns Serviços da AWS. Geralmente, esses são exemplos mais detalhados do que os exemplos do [guia de AWS CLI referência AWS CLI](#).
- [AWS CLI exemplos de comandos - exemplos](#) de comandos de código aberto que também estão disponíveis no [guia de AWS CLI referência](#) da . Os exemplos de comandos estão hospedados no [AWS CLI](#) repositório em. GitHub
- [AWS CLI usando exemplos de código de script Bash - exemplos](#) de script bash de código aberto. Os exemplos de scripts do Bash estão hospedados no [Repositório de exemplos AWS de código](#) em. GitHub

Exemplo de feedback

Você não consegue encontrar o que precisa? Solicite um exemplo de comando usando o link Fornecer feedback na parte inferior desta página ou na página de comando relevante no guia de [AWS CLI referência AWS CLI versão 2 do guia](#).

Quer contribuir? Contribua com exemplos de AWS CLI comandos no [Repositório de exemplos de AWS código](#) em. GitHub Para obter mais informações sobre como contribuir, consulte [exemplos de AWS CLI código: etapas rápidas de contribuição](#) nas GitHub páginas.

Exemplos de AWS CLI comandos guiados

O AWS Command Line Interface (AWS CLI) é uma ferramenta de código aberto que permite que você interaja com uma ampla variedade de comandos de Serviços da AWS uso em seu shell de linha de comando. Esta seção fornece exemplos guiados que mostram como aproveitar o AWS CLI para acessar alguns dos Serviços da AWS. Isso inclui alguns dos AWS CLI comandos personalizados, como os `aws s3` comandos de alto nível. Esses exemplos de comando demonstram ações comuns usadas por alguns Serviços da AWS e fornecem recursos adicionais para obter mais informações.

Se você é um AWS usuário experiente ou novo no AWS CLI, esses exemplos guiados servem como um recurso para simplificar suas AWS operações.

Para obter uma referência completa de todos os comandos disponíveis para cada um AWS service (Serviço da AWS), consulte o guia de [AWS CLI referência da AWS CLI versão 2 do guia](#) . Além disso, você pode utilizar a [ajuda integrada da linha de comando](#) para explorar a variedade de comandos Serviços da AWS, opções e recursos do AWS CLI.

Para obter mais exemplos de comandos que não estão disponíveis nesta seção, consulte a [AWS CLI exemplos de comando](#) seção. Esses são exemplos de comandos de código aberto que também estão disponíveis no guia de [AWS CLI referência da AWS CLI versão 2 do guia](#) . Os exemplos de comandos estão hospedados no [AWS CLI](#) repositório em. GitHub

Para exemplos de scripts bash de código aberto, consulte. [the section called “Exemplos de script Bash”](#) Os exemplos de scripts do Bash estão hospedados no [Repositório de exemplos AWS de código](#) em. GitHub

Serviços

- [Use o Amazon DynamoDB com o AWS CLI](#)
- [Use a Amazon EC2 com o AWS CLI](#)
- [Use o Amazon S3 Glacier com o AWS CLI](#)
- [Use AWS Identity and Access Management a partir do AWS CLI](#)
- [Use o Amazon S3 com o AWS CLI](#)
- [Uso do Amazon SNS com a AWS CLI](#)

Use o Amazon DynamoDB com o AWS CLI

Introdução ao Amazon DynamoDB

[O que é o Amazon DynamoDB?](#)

O AWS Command Line Interface (AWS CLI) fornece suporte para todos os serviços de AWS banco de dados, incluindo o Amazon DynamoDB. Você pode usar o AWS CLI para operações improvisadas, como criar uma tabela. Ela também pode ser usada para incorporar operações do DynamoDB em scripts utilitários.

Para obter mais informações sobre como usar o AWS CLI com o DynamoDB, [dynamodb](#) consulte na AWS CLI Referência de comandos.

Para listar os AWS CLI comandos do DynamoDB, use o comando a seguir.

```
$ aws dynamodb help
```

Tópicos

- [Pré-requisitos](#)
- [Criação e uso de tabelas do DynamoDB](#)
- [Uso do DynamoDB Local](#)
- [Recursos](#)

Pré-requisitos

Para executar os comandos dynamodb, você precisa:

- Instale e configure a AWS CLI. Para ter mais informações, consulte [Instalar a AWS CLI](#) e [Autenticação e credenciais de acesso](#).

Criação e uso de tabelas do DynamoDB

O formato da linha de comando consiste em um nome de comando do DynamoDB, seguido pelos parâmetros desse comando. O AWS CLI suporta a CLI [sintaxe abreviada](#) para os valores dos parâmetros e completa. JSON

O exemplo a seguir cria uma tabela chamada 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
```

Adicione novas linhas à tabela com comandos semelhantes aos mostrados no exemplo a seguir. Esses exemplos usam uma combinação de sintaxe abreviada 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"  
  }  
}
```

Pode ser difícil compor um comando válido JSON em uma única linha de comando. Para facilitar isso, eles AWS CLI podem ler JSON arquivos. Por exemplo, considere o seguinte JSON trecho, que é armazenado em um arquivo chamado. `expression-attributes.json`

```
{  
  ":v1": {"S": "No One You Know"},  
  ":v2": {"S": "Call Me Today"}  
}
```

Você pode usar esse arquivo para emitir uma solicitação query usando a AWS CLI. No exemplo a seguir, o conteúdo do arquivo `expression-attributes.json` é usado como o valor para o parâmetro `--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  
}
```

Uso do DynamoDB Local

Além do DynamoDB, você pode usar o AWS CLI com o DynamoDB Local. O DynamoDB Local é um banco de dados e servidor pequeno no lado do cliente que copia o serviço do DynamoDB. O DynamoDB Local permite que você escreva aplicativos que usam o DynamoDB, sem manipular tabelas ou dados no serviço web do API DynamoDB. Em vez disso, todas as API ações são redirecionadas para um banco de dados local. Isso economiza a taxa de transferência provisionada, o armazenamento de dados e as taxas de transferência de dados.

Para obter mais informações sobre o DynamoDB Local e como usá-lo com AWS CLI o, consulte as seguintes seções do Amazon [DynamoDB Developer Guide](#):

- [DynamoDB Local](#)
- [Como usar a AWS CLI com o DynamoDB Local](#)

Recursos

AWS CLI referência:

- [aws dynamodb](#)
- [aws dynamodb create-table](#)
- [aws dynamodb put-item](#)
- [aws dynamodb query](#)

Referência do serviço:

- [DynamoDB Local](#) no Guia do desenvolvedor do Amazon DynamoDB
- [Como usar a AWS CLI com o DynamoDB Local](#) no Guia do desenvolvedor do Amazon DynamoDB

Use a Amazon EC2 com o AWS CLI

Introdução ao Amazon Elastic Compute Cloud

[Introdução à Amazon EC2 - Elastic Cloud Server e hospedagem com AWS](#)

O Amazon Elastic Compute Cloud (AmazonEC2) fornece ambientes de computação virtual altamente escaláveis e flexíveis. A Amazon EC2 permite que você provisione e gerencie servidores virtuais, conhecidos como EC2 instâncias da Amazon, para atender a uma ampla variedade de necessidades de computação.

As EC2 instâncias da Amazon são máquinas virtuais que podem ser personalizadas com várias configurações de memóriaCPU, armazenamento e recursos de rede. Você pode escolher entre uma seleção diversificada de tipos de instância, desde opções leves e econômicas até instâncias poderosas e de alto desempenho, dependendo dos requisitos do seu aplicativo. Essa flexibilidade permite que você atenda às suas necessidades de computação para otimizar o desempenho e a economia.

Além disso, a Amazon EC2 oferece um conjunto de recursos que permite que você gerencie seus recursos computacionais de forma eficaz. Isso inclui a capacidade de iniciar rapidamente novas instâncias, criar imagens de máquina personalizadas (AMIs) para implantação rápida e aumentar ou diminuir a capacidade de computação conforme necessário.

Você pode acessar os recursos da Amazon EC2 usando o AWS Command Line Interface (AWS CLI). Para listar os AWS CLI comandos para a AmazonEC2, use o comando a seguir.

```
aws ec2 help
```

Antes de executar quaisquer comandos, defina suas credenciais padrão. Para obter mais informações, consulte [Configurar o AWS CLI](#).

Este tópico mostra exemplos resumidos de AWS CLI comandos que realizam tarefas comuns para a AmazonEC2.

Para exemplos longos de AWS CLI comandos, consulte o [repositório de exemplos de AWS CLI código](#) em. GitHub

Tópicos

- [Crie, exiba e exclua pares de EC2 chaves da Amazon no AWS CLI](#)
- [Criar, configurar e excluir grupos de segurança para o Amazon EC2](#)
- [Inicie, liste e feche EC2 instâncias da Amazon para o AWS CLI](#)
- [Alterar um tipo de instância do Amazon EC2 com um script bash](#)

Crie, exiba e exclua pares de EC2 chaves da Amazon no AWS CLI

Você pode usar o AWS Command Line Interface (AWS CLI) para criar, exibir e excluir seus pares de chaves para o Amazon Elastic Compute Cloud (AmazonEC2). Você usa pares de chaves para se conectar a uma EC2 instância da Amazon.

Você deve fornecer o par de chaves para a Amazon EC2 ao criar a instância e, em seguida, usar esse par de chaves para autenticar quando você se conectar à instância.

Note

Para obter exemplos adicionais de comandos, consulte o [guia AWS CLI de referência do guia](#) de .

Tópicos

- [Pré-requisitos](#)

- [Criar um par de chaves](#)
- [Exibir o par de chaves](#)
- [Excluir o par de chaves](#)
- [Referências](#)

Pré-requisitos

Para executar os comandos `ec2`, você precisa:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI e Autenticação e credenciais de acesso](#).
- Defina suas IAM permissões para permitir o EC2 acesso à Amazon. Para obter mais informações sobre IAM permissões para a AmazonEC2, consulte [IAMas políticas da Amazon EC2](#) no Guia EC2 do usuário da Amazon.

Criar um par de chaves

Para criar um par de chaves, use o comando `aws ec2 create-key-pair` com a opção `--query` e a opção `--output text` para canalizar sua chave privada diretamente para um arquivo.

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
> MyKeyPair.pem
```

Para PowerShell, o `> file` redirecionamento usa como padrão a codificação UTF -8, que não pode ser usada com alguns clientes. SSH Portanto, você deve converter a saída redirecionando-a para o comando `out-file` e definir explicitamente a codificação para `ascii`.

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text  
| out-file -encoding ascii -filepath MyKeyPair.pem
```

O arquivo resultante `MyKeyPair.pem` é semelhante ao seguinte.

```
-----BEGIN RSA PRIVATE KEY-----  
EXAMPLEKEYKCAQEAY7WZhaDsR1W3mR1QtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/  
vBoU7jLxx92pNHofnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7i05dSrvC7dQkW2duV5QuUdE0QW  
Z/aNxMniGQE6XAgfwlnXVBwreerrQo+ZwQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449i1x9X1F  
G50TCFe0zf18dqqCP6GzbPaIjiU19xX/az0R9V+tpU0zEL+wmXnZt3/nHPQ5xvD20JH67km6SuPW  
oPzev/D8V+x4+bHthfSjR9Y7DvQFjfbVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrrqu
```

```

/uler7vgIn5m71N5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/ufGxbl1
mb5qwMGUnEpJaZD6QSSs3kICLwUYUiGfc0uiSbmJoap/GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2
bahyWyJNfjLe4M86yd2YK3V2CmK+X/B0sShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/tJWSD9
81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA30zdXzMQexXVJ1TLZVEH0E7bh1Y9d801ozR
oQs/FiZNAx2iijCwyv01pjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfq1+1Ip1
YkriL0DbLXLvRAH+yHPRit2hH0jtUNZh4Axv+cpq09qbUI3+43eEy24B7G/Uh+GTfbjsXs0xQx/x
p9otyVwc7hsQ5TA5PZb+mvkJ50BEKzet9XcKw0NBYELGhnEPe7cCgYEA06Vgov6YH1eHui9kHuws
ayav0e1c5zKxjF9nfHFJRry21R1trw2Vdpn+9g481URrpzWV0Eihvm+xTtmaZlSp//lkq75XDwnU
WA8gkn603QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Un12ajLivWUt5pbBrKbUC
gYBjb0+0Zk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9Pfn295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY70zd5wQewBQ4AdS1WSX4nGDtsiFxiI5sKuAAe0CbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vwwKBgF+09VI/1wJBirsDGz9whVwFFPrTkJNvJZzYt69qezx1sjgFKshy
WBhd4xHZtmCqpBP1AymEjr/T01bxyARmXMnIOWIANNXMGB4KGSy11mzSVAoQ+fqR+cJ3d0dyP11j
jjb0Ed/NY8frlNDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0i0egLDa
NWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvTW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asinF3qS
VRkAKKKYeGjKpUfVTTrW0YFjXkfcR/V+QFL50ndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----

```

Sua chave privada não está armazenada AWS e só pode ser recuperada quando é criada. Não será possível recuperá-la posteriormente. Ao invés disso, se você perder a chave privada, deverá criar um novo par de chaves.

Se você estiver se conectando à sua instância a partir de um computador com Linux, recomendamos que você use o seguinte comando para definir as permissões do arquivo de chave privada, de maneira que apenas você possa lê-lo.

```
$ chmod 400 MyKeyPair.pem
```

Exibir o par de chaves

A “impressão digital” é gerada a partir do par de chaves, e você pode usá-la para verificar se a chave privada presente em sua máquina local corresponde à chave pública armazenada na AWS.

A impressão digital é um SHA1 hash retirado de uma cópia DER codificada da chave privada. Esse valor é capturado quando o par de chaves é criado e armazenado AWS com a chave pública. Você pode visualizar a impressão digital no EC2 console da Amazon ou executando o AWS CLI comando [aws ec2 describe-key-pairs](#).

O exemplo a seguir exibe a impressão digital para 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"  
  }  
]  
}
```

Para obter mais informações sobre chaves e impressões digitais, consulte [Amazon EC2 Key Pairs](#) no Guia do EC2 usuário da Amazon.

Excluir o par de chaves

Para excluir um par de chaves, execute o [aws ec2 delete-key-pair](#) comando, substituindo *MyKeyPair* com o nome do par a ser excluído.

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

Referências

AWS CLI referência:

- [aws ec2](#)
- [aws ec2 create-key-pair](#)
- [aws ec2 delete-key-pair](#)
- [aws ec2 describe-key-pairs](#)

Outra referência:

- [Documentação do Amazon Elastic Compute Cloud](#)
- Para ver, contribuir AWS SDK e AWS CLI codificar exemplos, consulte o [Repositório AWS de exemplos de código](#) em GitHub.

Criar, configurar e excluir grupos de segurança para o Amazon EC2

Você pode criar um grupo de segurança para suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) que atua essencialmente como um firewall, com regras que determinam qual tráfego de rede pode entrar e sair.

Use o AWS Command Line Interface (AWS CLI) para criar um grupo de segurança, adicionar regras aos grupos de segurança existentes e excluir grupos de segurança.

Note

Para obter exemplos adicionais de comandos, consulte o [guia AWS CLI de referência do guia](#) de .

Tópicos

- [Pré-requisitos](#)
- [Criar um grupo de segurança](#)
- [Adicionar regras ao grupo de segurança](#)
- [Excluir o grupo de segurança](#)
- [Referências](#)

Pré-requisitos

Para executar os comandos ec2, você precisa:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI e Autenticação e credenciais de acesso](#).
- Defina suas permissões do IAM para permitir acesso ao Amazon EC2. Para obter mais informações sobre as permissões do IAM para o Amazon EC2, consulte [as políticas do IAM para o Amazon EC2](#) no Guia do usuário do Amazon EC2.

Criar um grupo de segurança

É possível criar grupos de segurança associados a nuvens privadas virtuais (VPCs).

O exemplo do [aws ec2 create-security-group](#) a seguir mostra como criar um grupo de segurança para a VPC especificada.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --  
vpc-id vpc-1a2b3c4d  
{  
  "GroupId": "sg-903004f8"
```

```
}
```

Para visualizar as informações iniciais para um grupo de segurança, execute o comando [aws ec2 describe-security-groups](#). Você pode fazer referência a um grupo de segurança do EC2-VPC somente por seu `vpc-id`, e não apenas por seu 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"
    }
  ]
}
```

Adicionar regras ao grupo de segurança

Quando você executa uma instância do Amazon EC2, é necessário ativar regras no grupo de segurança para permitir o tráfego de rede de entrada como meio de se conectar à imagem.

Por exemplo, se estiver iniciando uma instância do Windows, normalmente você adiciona uma regra para permitir que o tráfego de entrada na porta TCP 3389 ofereça suporte ao Protocolo de Desktop Remoto (RDP). Se estiver iniciando uma instância do Linux, geralmente você adiciona uma regra para permitir que o tráfego de entrada na porta 22 ofereça suporte às conexões SSH.

Use o comando [aws ec2 authorize-security-group-ingress](#) para adicionar uma regra a um grupo de segurança. Um parâmetro obrigatório deste comando é o endereço IP público de seu computador ou rede (na forma de um intervalo de endereços) ao qual seu computador está conectado na notação [CIDR](#).

Note

Fornecemos o serviço <https://checkip.amazonaws.com/> para permitir a você determinar seu endereço IP público. Para encontrar outros serviços que podem ajudar você a identificar o endereço IP, use o navegador para pesquisar “qual é meu endereço IP”. Se você se conectar por meio de um ISP ou protegido por um firewall usando um endereço IP dinâmico (por meio de um gateway NAT de uma rede privada), seu endereço poderá mudar periodicamente. Nesse caso, você deve descobrir o intervalo de endereços IP usados por computadores cliente.

O exemplo a seguir mostra como adicionar uma regra para RDP (porta TCP 3389) em um grupo de segurança do EC2-VPC com o ID `sg-903004f8` usando seu endereço IP.

Para começar, encontre seu endereço IP.

```
$ curl https://checkip.amazonaws.com
x.x.x.x
```

Em seguida, você pode adicionar o endereço IP ao seu grupo de segurança, executando o comando [aws ec2 authorize-security-group-ingress](#).

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 3389 --cidr x.x.x.x/x
```

O comando a seguir adiciona uma regra para habilitar o SSH para instâncias no mesmo grupo de segurança.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 22 --cidr x.x.x.x/x
```

Para visualizar as alterações no grupo de segurança, execute o comando [aws ec2 describe-security-groups](#).

```
$ 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"
    }
  ]
}
```

Excluir o grupo de segurança

Para excluir um grupo de segurança, execute o comando [aws ec2 delete-security-group](#).

Note

Não é possível excluir um grupo de segurança se ele estiver atualmente anexado a um ambiente.

O exemplo de comando a seguir exclui um grupo de segurança do EC2-VPC.

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

Referências

AWS CLI referência:

- [aws ec2](#)
- [aws ec2 authorize-security-group-ingress](#)
- [aws ec2 create-security-group](#)
- [aws ec2 delete-security-group](#)
- [aws ec2 describe-security-groups](#)

Outra referência:

- [Documentação do Amazon Elastic Compute Cloud](#)
- Para ver e contribuir com exemplos de AWS SDK e AWS CLI código, consulte o [Repositório de exemplos de AWS código](#) em GitHub

Inicie, liste e feche EC2 instâncias da Amazon para o AWS CLI

Você pode usar o AWS Command Line Interface (AWS CLI) para iniciar, listar e encerrar instâncias do Amazon Elastic Compute Cloud EC2 (Amazon). Se você iniciar uma instância que não esteja dentro do nível AWS gratuito, você será cobrado após a execução da instância e cobrado pelo tempo em que a instância estiver em execução, mesmo que ela permaneça ociosa.

Note

Para obter exemplos adicionais de comandos, consulte o [guia AWS CLI de referência do guia](#) de .

Tópicos

- [Pré-requisitos](#)
- [Executar sua instância](#)
- [Adicionar um dispositivo de blocos à instância](#)
- [Adicionar uma etiqueta à instância](#)
- [Conecte-se à sua instância](#)
- [Listar as instâncias](#)
- [Encerrar a instância](#)
- [Referências](#)

Pré-requisitos

Para executar os comandos ec2 neste tópico, você precisa:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI e Autenticação e credenciais de acesso](#).
- Defina suas IAM permissões para permitir o EC2 acesso à Amazon. Para obter mais informações sobre IAM permissões para a AmazonEC2, consulte [IAMas políticas da Amazon EC2](#) no Guia EC2 do usuário da Amazon.
- Crie um [par de chaves](#) e um [grupo de segurança](#).
- Selecione uma imagem de máquina da Amazon (AMI) e anote o AMI ID. Para obter mais informações, consulte [Encontrar um adequado AMI](#) no Guia do EC2 usuário da Amazon.

Executar sua instância

Para iniciar uma EC2 instância da Amazon usando o AMI que você selecionou, use o [aws ec2 run-instances](#) comando. Você pode iniciar a instância em uma nuvem privada virtual (VPC).

Inicialmente, a instância será exibida no estado `pending`, mas mudará para o estado `running` depois de alguns minutos.

O exemplo a seguir mostra como iniciar uma `t2.micro` instância na sub-rede especificada de a VPC Substitua o *italicized* valores de parâmetros com os seus.

```
$ 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
}
]
```

Adicionar um dispositivo de blocos à instância

Cada instância lançada tem um volume do dispositivo root associados. Você pode usar o mapeamento de dispositivos de blocos para especificar volumes adicionais do Amazon Elastic Block Store (AmazonEBS) ou volumes de armazenamento de instâncias a serem anexados a uma instância quando ela for executada.

Para adicionar um dispositivo de blocos em sua instância, especifique a opção `--block-device-mappings` ao usar `run-instances`.

O parâmetro de exemplo a seguir provisiona um EBS volume padrão da Amazon com 20 GB de tamanho e o mapeia para sua instância usando o identificador `/dev/sdf`.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\", \"Ebs\":{\"VolumeSize\":20, \"DeleteOnTermination\":false} }]"
```

O exemplo a seguir adiciona um EBS volume da Amazon, mapeado para `/dev/sdf`, com base em um snapshot existente. Um snapshot representa uma imagem que é carregada no volume para você. Ao especificar um snapshot, não é necessário especificar um tamanho de volume; ele será grande o suficiente para armazenar sua imagem. No entanto, se você especificar um tamanho, ele deverá ser maior que ou igual ao tamanho do snapshot.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\", \"Ebs\":{\"SnapshotId\":\"snap-a1b2c3d4\"} }]"
```

O exemplo a seguir adiciona dois volumes à sua instância. O número de volumes disponíveis para sua instância depende do seu tipo de instância.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\", \"VirtualName\":\"ephemeral0\"}, {\"DeviceName\":\"/dev/sdg\", \"VirtualName\":\"ephemeral1\"}]"
```

O exemplo a seguir cria o mapeamento (`/dev/sdj`), mas não provisiona um volume para a instância.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdj\", \"NoDevice\":\"\"}]"
```

Para obter mais informações, consulte [Mapeamento de dispositivos de blocos](#) no Guia EC2 do usuário da Amazon.

Adicionar uma etiqueta à instância

Uma tag é um rótulo que você atribui a um AWS recurso. Permite adicionar metadados aos seus recursos que você pode usar para diversas finalidades. Para obter mais informações, consulte [Como marcar seus recursos](#) no Guia do EC2 usuário da Amazon.

O exemplo a seguir mostra como adicionar uma etiqueta com o nome da chave “Name” e o valor “MyInstance” à instância especificada com o comando [aws ec2 create-tags](#).

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

Conecte-se à sua instância

Durante a execução da instância, é possível se conectar a ela e usá-la da mesma forma que você usaria um computador. Para obter mais informações, consulte [Connect to Your Amazon EC2 Instance](#) no Guia EC2 do usuário da Amazon.

Listar as instâncias

Você pode usar o AWS CLI para listar suas instâncias e visualizar informações sobre elas. Liste todas as suas instâncias, ou filtre os resultados de acordo com as instâncias de interesse.

Os exemplos a seguir mostram como usar o comando [aws ec2 describe-instances](#).

O comando a seguir relaciona todas as suas instâncias.

```
$ aws ec2 describe-instances
```

O comando a seguir filtra a lista apenas para suas instâncias t2.micro e mostra apenas os valores InstanceId para cada correspondência.

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query
"Reservations[].Instances[].InstanceId"
[
  "i-05e998023d9c69f9a"
]
```

O comando a seguir lista todas as suas instâncias que tem a etiqueta Name=MyInstance.

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

O comando a seguir lista suas instâncias que foram executadas usando qualquer um dos seguintes AMIs: ami-x0123456ami-y0123456, ami-z0123456 e.

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-
y0123456,ami-z0123456"
```

Encerrar a instância

O encerramento de uma instância significa excluí-la. Você não pode se reconectar com uma instância depois de tê-la encerrado.

Assim que o estado da instância de mudar para shutting-down ou para terminated, não há mais custos para essa instância. Se você desejar se reconectar a ela mais tarde, use [stop-instances](#) em vez de `terminate-instances`. Para obter mais informações, consulte [Encerre sua instância](#) no Guia do EC2 usuário da Amazon.

Para excluir uma instância, use o comando [aws ec2 terminate-instances](#).

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-5203422c",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

Referências

AWS CLI referência:

- [aws ec2](#)
- [aws ec2 create-tags](#)
- [aws ec2 describe-instances](#)
- [aws ec2 run-instances](#)
- [aws ec2 terminate-instances](#)

Outra referência:

- [Documentação do Amazon Elastic Compute Cloud](#)
- Para ver, contribuir AWS SDK e AWS CLI codificar exemplos, consulte o [Repositório AWS de exemplos de código](#) em GitHub.

Alterar um tipo de instância do Amazon EC2 com um script bash

Este exemplo de script bash para o Amazon EC2 altera o tipo de instância de uma instância do Amazon EC2 usando o (). AWS Command Line Interface AWS CLI Ele interrompe a instância se ela estiver em execução, altera o tipo de instância e, em seguida, se solicitado, a reinicia. Scripts shell são programas desenvolvidos para ser executados em uma interface de linha de comando.

Note

Para obter exemplos adicionais de comandos, consulte o [guia AWS CLI de referência do guia](#) de .

Tópicos

- [Antes de começar](#)
- [Sobre este exemplo](#)
- [Parâmetros](#)
- [Arquivos](#)
- [Referências](#)

Antes de começar

Antes que você possa executar qualquer um dos exemplos abaixo, as seguintes tarefas deverão ser concluídas.

- Instale e configure a AWS CLI. Para ter mais informações, consulte [Instalar a AWS CLI e Autenticação e credenciais de acesso](#).
- O perfil que você usa deve ter permissões que permitam as AWS operações executadas pelos exemplos.
- Uma instância do Amazon EC2 em execução na conta para a qual você tem permissão para interromper e modificar. Se você executar o script de teste, ele iniciará uma instância para você, testará alterando o tipo e encerrará a instância.
- Como prática AWS recomendada, conceda a esse código o menor privilégio ou somente as permissões necessárias para realizar uma tarefa. Para obter mais informações, consulte [Conceder privilégio mínimo](#) no Manual do usuário do AWS Identity and Access Management (IAM).

- Esse código não foi testado em todas as AWS regiões. Alguns AWS serviços estão disponíveis somente em regiões específicas. Para obter mais informações, consulte [Endpoints de serviço e cotas](#) no Guia de referência geral da AWS .
- A execução desse código pode resultar em cobranças em sua AWS conta. É sua responsabilidade garantir que todos os recursos criados por este script sejam removidos após você terminar de usá-los.

Sobre este exemplo

Este exemplo é escrito como uma função no arquivo de script shell `change_ec2_instance_type.sh` que você pode usar como `source` a partir de outro script ou da linha de comando. Cada arquivo de script contém comentários descrevendo cada uma das funções. Uma vez que a função esteja na memória, você poderá chamá-la via linha de comando. Por exemplo, os seguintes comandos alteram o tipo de instância especificada para `t2.nano`:

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

Para ver o exemplo completo e os arquivos de script disponíveis para download, consulte [Alterar o tipo de instância do Amazon EC2](#) no repositório de exemplos de código em AWS . GitHub

Parâmetros

`-i`: (string) especifica o ID da instância a ser modificada.

`-t`: (string) especifica o tipo de instância do Amazon EC2 de destino.

`-r`: (opção) por padrão, não está definida. Se `-r` estiver definido, reiniciará a instância após a alteração do tipo.

`-f`: (opção) por padrão, o script solicita ao usuário que confirme o encerramento da instância antes de fazer a mudança. Se `-f` estiver definida, a função não avisará o usuário antes de encerrar a instância para realizar a alteração de tipo.

`-v`: (opção) por padrão, o script opera silenciosamente e exibe a saída somente em caso de erro. Se `-v` estiver definida, a função exibirá o status ao longo de sua operação.

Arquivos

change_ec2_instance_type.sh

O arquivo de script principal contém a função `change_ec2_instance_type()` que executa as seguintes tarefas:

- Verifica se a instância do Amazon EC2 especificada existe.
- A menos que a opção `-f` esteja selecionada, avisará o usuário antes de interromper a instância.
- Altera o tipo de instância
- Se você definir `-r`, reiniciará a instância e confirmará que a instância está sendo executada

Veja o código para [change_ec2_instance_type.sh](#) on GitHub.

test_change_ec2_instance_type.sh

O script do arquivo `test_change_ec2_instance_type.sh` testa os vários caminhos de código para a função `change_ec2_instance_type`. Se todas as etapas no script de teste funcionarem corretamente, ele removerá todos os recursos que criou.

Você pode executar o script de teste com os seguintes parâmetros:

- `-v`: (opção) cada teste mostra um status de aprovação/falha conforme eles são executados. Por padrão, os testes são executados silenciosamente e a saída inclui apenas o status final de aprovação/falha.
- `-i`: (opção) o script faz uma pausa após cada teste para permitir que você navegue pelos resultados intermediários de cada etapa. Isso permite examinar o status atual da instância usando o console do Amazon EC2. O script avançará para a próxima etapa após você pressionar ENTER no prompt.

Veja o código para [test_change_ec2_instance_type.sh](#) on GitHub.

awsdocs_general.sh

O arquivo de script `awsdocs_general.sh` contém funções de uso geral usadas em exemplos avançados para a AWS CLI.

Veja o código para [awsdocs_general.sh](#) on GitHub.

Referências

AWS CLI referência:

- [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](#)

Outra referência:

- [Documentação do Amazon Elastic Compute Cloud](#)
- Para ver e contribuir com exemplos de AWS SDK e AWS CLI código, consulte o [Repositório de exemplos de AWS código](#) em. GitHub

Use o Amazon S3 Glacier com o AWS CLI

Introdução ao Amazon S3 Glacier

[Introdução ao Amazon S3 Glacier](#)

Este tópico mostra exemplos de AWS CLI comandos que realizam tarefas comuns para o S3 Glacier. Os exemplos demonstram como usar o AWS CLI para fazer upload de um arquivo grande para o S3 Glacier dividindo-o em partes menores e enviando-as a partir da linha de comando.

Você pode acessar os recursos do Amazon S3 Glacier usando o AWS Command Line Interface ().AWS CLI Para listar os AWS CLI comandos do S3 Glacier, use o comando a seguir.

```
aws glacier help
```

Note

Para obter referência de comandos e exemplos adicionais, consulte [aws glacier](#) na Referência de comandos da AWS CLI .

Tópicos

- [Pré-requisitos](#)
- [Criar um cofre do Amazon S3 Glacier](#)
- [Preparar um arquivo para upload](#)
- [Iniciar um multipart upload e fazer upload de arquivos](#)
- [Concluir o upload](#)
- [Recursos](#)

Pré-requisitos

Para executar os comandos `glacier`, você precisa:

- Instale e configure a AWS CLI. Para ter mais informações, consulte [Instalar a AWS CLI](#) e [Autenticação e credenciais de acesso](#).
- Este tutorial usa várias ferramentas de linha de comando que normalmente vêm pré-instaladas em sistemas operacionais Unix, incluindo Linux e macOS. Os usuários do Windows podem usar as mesmas ferramentas instalando o [Cygwin](#) e executando os comandos do terminal Cygwin. São observados comandos originários do Windows e utilitários que executam as mesmas funções onde disponível.

Criar um cofre do Amazon S3 Glacier

Crie um cofre com o [create-vault](#).

```
$ aws glacier create-vault --account-id - --vault-name myvault
{
  "location": "/123456789012/vaults/myvault"
}
```

Note

Todos os comandos do S3 Glacier exigem um parâmetro de ID de conta. Use o caractere de hífen (`--account-id -`) para usar a conta atual.

Preparar um arquivo para upload

Crie um arquivo para o upload de teste. Os comandos a seguir criam um arquivo chamado *largefile* que contém exatamente 3 MiB de dados aleatórios.

Linux ou 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 é um utilitário que copia um número de bytes a partir de um arquivo de entrada para um arquivo de saída. O exemplo anterior usa o arquivo de dispositivo do sistema `/dev/urandom` como uma fonte de dados aleatórios. O `fsutil` executa uma função semelhante no Windows.

Windows

```
C:\> fsutil file createnew largefile 3145728
File C:\temp\largefile is created
```

Em seguida, divida o arquivo em pedaços de 1 MiB (1.048.576 bytes) usando um divisor de arquivos.

```
$ split -b 1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

Iniciar um multipart upload e fazer upload de arquivos

Crie um upload fracionado no Amazon S3 Glacier usando o 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_lR7vgFuJV6NtcV5zpsJ",
  "location": "/123456789012/vaults/myvault/multipart-uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
}
```

O S3 Glacier exige o tamanho de cada parte em bytes (1 MiB neste exemplo), o nome do cofre e uma ID de conta para configurar o upload fracionado. A AWS CLI saída é uma ID de upload quando a operação é concluída. Salve o ID de upload para um shell variável para uso posterior.

Linux ou macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

Depois, use o comando [upload-multipart-part](#) para fazer upload das três partes.

```
$ 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

O exemplo anterior usa o sinal de dólar (\$) para fazer referência ao conteúdo da variável de shell UPLOADID no Linux. Na linha de comando do Windows, use um sinal de porcentagem (%) nos dois lados do nome da variável (por exemplo, %UPLOADID%).

Especifique o intervalo de bytes de cada parte ao fazer upload para que o S3 Glacier possa remontá-lo na ordem correta. Cada parte é 1.048.576 bytes, portanto, a primeira parte ocupa 0-1048575 bytes, a segunda 1048576-2097151 e a terceira 2097152-3145727.

Concluir o upload

O Amazon S3 Glacier exige um hash de árvore do arquivo original para confirmar que todas as partes enviadas chegaram intactas. AWS

Para calcular um hash de árvore, você deve dividir o arquivo em partes de 1 MiB e calcular um hash SHA binário -256 de cada peça. Em seguida, divida a lista de hashes em pares, combine os dois binários hashes em cada par e execute hashes dos resultados. Repita esse processo até que haja apenas um hash à esquerda. Se houver um número ímpar de hashes em qualquer nível, envie para o próximo nível sem modificá-lo.

A chave para calcular uma árvore hash corretamente ao usar os utilitários de linha de comando é armazenar cada hash em formato binário e converter apenas para hexadecimal na última etapa. A combinação ou hash de qualquer versão hexadecimal hash em árvore gerará um resultado incorreto.

Note

Os usuários do Windows podem usar o comando type em vez do cat. SSLO Open está disponível para Windows em SSLOpen.org.

Para calcular uma árvore hash

1. Se ainda não fez isso, divida o arquivo original em partes de 1 MiB.

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
```

```
creating file `chunkac`
```

2. Calcule e armazene o hash binário SHA -256 de cada bloco.

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. Combine os primeiros dois hashes e execute o hash binário do resultado.

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. Combine o pai de partes de hash aa e ab com o hash de bloco ac e o resultado do hash, desta vez exibindo hexadecimal. Armazene o resultado em um shell variável.

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

Por fim, preencha o upload com o comando [complete-multipart-upload](#). Este comando usa o tamanho do arquivo original em bytes, o valor de hash em árvore final em hexadecimal, e a ID da conta e nome do cofre.

```
$ aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728 --
upload-id $UPLOADID --account-id - --vault-name myvault
{
  "archiveId": "d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrALLGAA0NJAzo5QdP-
N83MKqd96Unspoa5H51ItWX-sK8-QS0ZhwsyGiu9-R-
kwWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
  "checksum": "9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
  "location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrALLGAA0NJAzo5QdP-N83MKqd96Unspoa5H51ItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

Também é possível verificar o status do cofre usando o 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

O status do cofre é atualizado cerca de uma vez por dia. Consulte [Como trabalhar com cofres](#) para obter mais informações.

Já é seguro remover o bloco e os arquivos de hash que você criou.

```
$ rm chunk* hash*
```

Para obter mais informações sobre uploads fracionados, consulte [Upload de arquivos grandes em partes](#) e [Como computar somas de verificação](#) no Guia do desenvolvedor do Amazon S3 Glacier.

Recursos

AWS CLI referência:

- [aws glacier](#)
- [aws glacier complete-multipart-upload](#)
- [aws glacier create-vault](#)
- [aws glacier describe-vault](#)
- [aws glacier initiate-multipart-upload](#)

Referência do serviço:

- [Guia do desenvolvedor do Amazon S3 Glacier](#)
- [Carregar arquivos grandes em partes](#) no Guia do desenvolvedor do Amazon S3 Glacier
- [Calcular somas de verificação](#) no Guia do desenvolvedor do Amazon S3 Glacier
- [Trabalhar com vaults](#) no Guia do desenvolvedor do Amazon S3 Glacier

Use AWS Identity and Access Management a partir do AWS CLI

Uma introdução ao AWS Identity and Access Management

[Introdução ao AWS Identity and Access Management](#)

Você pode acessar os recursos do AWS Identity and Access Management (IAM) usando o AWS Command Line Interface (AWS CLI). Para listar os AWS CLI comandos para IAM, use o comando a seguir.

```
aws iam help
```

Este tópico mostra exemplos de AWS CLI comandos que executam tarefas comuns para IAM.

Antes de executar quaisquer comandos, defina suas credenciais padrão. Para obter mais informações, consulte [Configurar o AWS CLI](#).

Para obter mais informações sobre o IAM serviço, consulte o [Guia AWS Identity and Access Management do usuário](#).

Tópicos

- [Criar usuários e grupos do IAM](#)
- [Anexar uma política IAM gerenciada a um usuário](#)
- [Definir uma senha inicial para um IAM usuário](#)
- [Criar uma chave de acesso para um IAM usuário](#)

Criar usuários e grupos do IAM

Como criar um grupo e adicionar um novo usuário a ele

1. Use o comando [create-group](#) para criar o grupo.

```
$ 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. Use o comando [create-user](#) para criar o usuário.

```
$ 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. Use o comando [add-user-to-group](#) para adicionar o usuário ao grupo.

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. Para verificar se o grupo MyIamGroup contém MyUser, use o 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"  
}
```

Anexar uma política IAM gerenciada a um usuário

Neste exemplo, a política fornece ao usuário “Acesso de Usuário Power”.

Para anexar uma política IAM gerenciada a um usuário

1. Determine o nome do recurso Amazon (ARN) da política a ser anexada. O comando a seguir `list-policies` é usado para encontrar a ARN política com o nome `PowerUserAccess`. Em seguida, ele armazena isso ARN em uma variável de ambiente.

```
$ export POLICYARN=$(aws iam list-policies --query 'Policies[?  
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text) ~  
$ echo $POLICYARN  
arn:aws:iam::aws:policy/PowerUserAccess
```

2. Para anexar a política, use o [attach-user-policy](#) comando e faça referência à variável de ambiente que contém a política ARN.

```
$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN
```

3. Verifique se a política foi anexada ao usuário executando o comando [list-attached-user-policies](#).

```
$ aws iam list-attached-user-policies --user-name MyUser  
{  
  "AttachedPolicies": [  
    {  
      "PolicyName": "PowerUserAccess",  
      "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Recursos de gerenciamento de acesso](#). Este tópico fornece links para uma visão geral de permissões e políticas e links para exemplos de políticas para acessar o Amazon S3EC2, a Amazon e outros serviços.

Definir uma senha inicial para um IAM usuário

O comando a seguir usa [create-login-profile](#) para definir uma senha inicial para o usuário especificado. Quando o usuário faz login pela primeira vez, ele precisa alterar a senha para algo que apenas ele sabe.

```
$ 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
  }
}
```

Você pode usar o comando `update-login-profile` para alterar a senha para um usuário.

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

Criar uma chave de acesso para um IAM usuário

Você pode usar o comando [create-access-key](#) para criar uma chave de acesso para um usuário. Uma chave de acesso é um conjunto de credenciais de segurança que consiste em um ID de chave de acesso e uma chave secreta.

Um usuário pode criar apenas duas chaves de acesso ao mesmo tempo. Se você tentar criar um terceiro conjunto, o comando retornará um erro `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"
  }
}
```

Use o comando [delete-access-key](#) para excluir uma chave de acesso para um usuário. Especifique qual chave de acesso deve ser excluída usando o ID de chave de acesso.

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

Use o Amazon S3 com o AWS CLI

Introdução ao Amazon Simple Storage Service (Amazon S3)

[Introdução ao Amazon Simple Storage Service \(Amazon S3\): armazenamento na nuvem na AWS](#)

Você pode acessar os recursos do Amazon Simple Storage Service (Amazon S3) usando AWS Command Line Interface o ().AWS CLI O Amazon S3 é um serviço de armazenamento de objetos altamente escalável e durável. O Amazon S3 foi projetado para fornecer capacidade de armazenamento praticamente ilimitada, tornando-o uma solução ideal para uma ampla variedade de necessidades de armazenamento e gerenciamento de dados.

O Amazon S3 permite que você armazene e recupere qualquer quantidade de dados, de pequenos arquivos a grandes conjuntos de dados, na forma de objetos. Cada objeto é armazenado em um contêiner chamado bucket, que pode ser acessado e gerenciado por meio do AWS Management Console ou programaticamente por meio das AWS SDKs ferramentas e. AWS CLI

Incluindo armazenamento básico, o Amazon S3 também oferece uma variedade de recursos, incluindo gerenciamento do ciclo de vida, controle de versão, escalabilidade e segurança. Eles se integram a outros, Serviços da AWS permitindo que você crie soluções baseadas em nuvem que se adaptam às suas necessidades.

AWS CLI Ele fornece dois níveis de comandos para acessar o Amazon S3:

- **s3** — Comandos personalizados de alto nível feitos especificamente para o AWS CLI que simplificam a execução de tarefas comuns, como criar, manipular, excluir e sincronizar objetos e buckets.
- **s3api** — Expõe acesso direto a todas as operações do Amazon S3, o que permite API realizar operações avançadas.

Tópicos neste guia:

- [Use comandos de alto nível \(s3\) com o AWS CLI](#)

- [Use os API comandos -Level \(s3api\) com o AWS CLI](#)
- [Exemplo de script de operações de ciclo de vida de bucket Amazon S3](#)

Use comandos de alto nível (s3) com o AWS CLI

Este tópico descreve alguns dos comandos que você pode utilizar para gerenciar buckets e objetos do Amazon S3 usando os comandos [aws s3](#) na AWS CLI. Para comandos não abordados neste tópico e exemplos de comandos adicionais, consulte os comandos [aws s3](#) na Referência da AWS CLI .

Os comandos `aws s3` de alto nível simplificam o gerenciamento de objetos do Amazon S3. Esses comandos permitem a você gerenciar o conteúdo do Amazon S3 dentro dele mesmo e com diretórios locais.

Tópicos

- [Pré-requisitos](#)
- [Antes de começar](#)
- [Criar um bucket](#)
- [Listar buckets e objetos](#)
- [Excluir buckets](#)
- [Excluir objetos](#)
- [Mover objetos](#)
- [Copiar objetos](#)
- [Sincronizar objetos](#)
- [Opções usadas com frequência para comandos s3](#)
- [Recursos](#)

Pré-requisitos

Para executar os comandos `s3`, você precisa:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI](#) e [Autenticação e credenciais de acesso](#).
- O perfil que você usa deve ter permissões que permitam as AWS operações executadas pelos exemplos.

- Entenda estes termos do Amazon S3:
 - Bucket: uma pasta de nível superior do Amazon S3.
 - Prefixo: uma pasta do Amazon S3 em um bucket.
 - Objeto: qualquer item hospedado em um bucket do Amazon S3.

Antes de começar

Esta seção descreve algumas coisas a serem observadas antes de usar comandos da `aws s3`.

Carregamentos de objetos grandes

Quando você usa comandos da `aws s3` para fazer upload de objetos grandes em um bucket do Amazon S3, a AWS CLI executa automaticamente um upload fracionado. Quando esses comandos da `aws s3` são usados, não é possível continuar uploads que falharam.

Se o upload de várias partes falhar devido a um tempo limite, ou se você tiver cancelado manualmente no AWS CLI, o upload será AWS CLI interrompido e os arquivos criados serão limpos. Esse processo pode levar alguns minutos.

Se o processo de upload fracionado ou limpeza for cancelado por um comando `kill` ou uma falha do sistema, os arquivos criados permanecerão no bucket do Amazon S3.

Propriedades e tags de arquivos em cópias com várias partes

Quando você usa a AWS CLI versão 1 dos comandos no `aws s3` namespace para copiar um arquivo de um local de bucket do Amazon S3 para outro local do bucket do Amazon S3, e essa operação [usa cópia em várias partes](#), nenhuma propriedade do arquivo do objeto de origem é copiada para o objeto de destino.

Criar um bucket

Use o comando `s3 mb` para criar um bucket. Os nomes dos buckets devem ser globalmente exclusivos (exclusivos em todo o Amazon S3) e devem estar DNS em conformidade.

Nomes de bucket podem conter letras minúsculas, números, hífen e pontos. Os nomes de bucket podem iniciar e terminar apenas com uma letra ou número e não podem conter um ponto ao lado de um hífen ou outro ponto.

Sintaxe


```
$ aws s3 mb <target> [--options]
```

Exemplos de s3 mb

O exemplo a seguir cria o bucket `s3://bucket-name`.

```
$ aws s3 mb s3://bucket-name
```

Listar buckets e objetos

Para listar seus buckets, pastas ou objetos, use o comando [s3 ls](#). O uso do comando sem um destino ou opções lista todos os buckets.

Sintaxe

```
$ aws s3 ls <target> [--options]
```

Para ver algumas opções comuns que podem ser usadas com este comando e exemplos, consulte [Opções usadas com frequência para comandos s3](#). Para obter uma lista completa de opções disponíveis, consulte [s3 ls](#) na Referência de comandos da AWS CLI .

Exemplos de s3 ls

O exemplo a seguir lista todos os seus buckets do Amazon S3.

```
$ aws s3 ls
2018-12-11 17:08:50 my-bucket
2018-12-14 14:55:44 my-bucket2
```

O comando a seguir lista todos os objetos e prefixos em um bucket. Neste exemplo de saída, o prefixo `example/` tem um arquivo chamado `MyFile1.txt`.

```
$ aws s3 ls s3://bucket-name
                                PRE example/
2018-12-04 19:05:48              3 MyFile1.txt
```

Você pode filtrar a saída para um prefixo específico, incluindo-o no comando. O comando a seguir lista os objetos em `bucket-name/example/` (ou seja, objetos em `bucket-name` filtrado pelo prefixo `example/`).

```
$ aws s3 ls s3://bucket-name/example/  
2018-12-06 18:59:32          3 MyFile1.txt
```

Excluir buckets

Para excluir um bucket, use o comando [s3 rb](#).

Sintaxe

```
$ aws s3 rb <target> [--options]
```

Exemplos de s3 rb

O exemplo a seguir remove o bucket `s3://bucket-name`.

```
$ aws s3 rb s3://bucket-name
```

Por padrão, o bucket deve estar vazio para a operação ser bem-sucedida. Para remover um bucket que não está vazio, é necessário incluir a opção `--force`. Se você estiver usando um bucket com versionamento que contém objetos anteriormente excluídos, mas ainda retidos, esse comando não permitirá que você remova o bucket. Você deve primeiro remover todo o conteúdo.

O exemplo de comando a seguir exclui todos os objetos no bucket e, em seguida, exclui o bucket.

```
$ aws s3 rb s3://bucket-name --force
```

Excluir objetos

Para excluir objetos em um bucket ou seu diretório local, use o comando [s3 rm](#).

Sintaxe

```
$ aws s3 rm <target> [--options]
```

Para ver algumas opções comuns que podem ser usadas com este comando e exemplos, consulte [Opções usadas com frequência para comandos s3](#). Para obter uma lista completa de opções, consulte [s3 rm](#) na Referência de comandos da AWS CLI .

Exemplos de s3 rm

O exemplo a seguir exclui `filename.txt` de `s3://bucket-name/example`.

```
$ aws s3 rm s3://bucket-name/example/filename.txt
```

O exemplo a seguir exclui todos os objetos de `s3://bucket-name/example` usando a opção `--recursive`.

```
$ aws s3 rm s3://bucket-name/example --recursive
```

Mover objetos

Use o comando [s3 mv](#) para mover objetos de um bucket ou diretório local. O `s3 mv` comando copia o objeto ou arquivo de origem para o destino especificado e, em seguida, exclui o objeto ou arquivo de origem.

Sintaxe

```
$ aws s3 mv <source> <target> [--options]
```

Para ver algumas opções comuns que podem ser usadas com este comando e exemplos, consulte [Opções usadas com frequência para comandos s3](#). Para obter uma lista completa de opções disponíveis, consulte [s3 mv](#) na Referência de comandos da AWS CLI .

Warning

Se você estiver usando qualquer tipo de ponto de acesso ARNs ou aliases de ponto de acesso em sua origem ou URIs destino do Amazon S3, você deve tomar cuidado extra para que seu Amazon S3 de origem e destino sejam resolvidos em diferentes URIs buckets subjacentes. Se os buckets de origem e destino forem iguais, o arquivo ou objeto de origem poderá ser movido para si mesmo, o que pode resultar na exclusão acidental do arquivo ou objeto de origem. Para verificar se os buckets de origem e destino não são os mesmos, use o `--validate-same-s3-paths` parâmetro ou defina [AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS](#) a true variável de ambiente como.

Exemplos de s3 mv

O exemplo a seguir move todos os objetos de `s3://bucket-name/example` para `s3://my-bucket/`.

```
$ aws s3 mv s3://bucket-name/example s3://my-bucket/
```

O exemplo a seguir move um arquivo local do diretório de trabalho atual para o bucket do Amazon S3 com o comando `s3 mv`.

```
$ aws s3 mv filename.txt s3://bucket-name
```

O exemplo a seguir move um arquivo do bucket do Amazon S3 para o diretório de trabalho atual, onde `./` especifica o diretório de trabalho atual.

```
$ aws s3 mv s3://bucket-name/filename.txt ./
```

Copiar objetos

Use o comando [s3 cp](#) para copiar objetos de um bucket ou diretório local.

Sintaxe

```
$ aws s3 cp <source> <target> [--options]
```

Você pode usar o parâmetro dash para transmitir arquivos para entrada padrão (`stdin`) ou saída padrão (`stdout`).

Warning

Se você estiver usando PowerShell, o shell pode alterar a codificação de a CRLF ou adicionar CRLF a à entrada ou saída canalizada ou à saída redirecionada.

O comando `s3 cp` usa a sintaxe a seguir para fazer upload de um fluxo de arquivos do `stdin` para um bucket especificado.

Sintaxe

```
$ aws s3 cp - <target> [--options]
```

O comando `s3 cp` usa a sintaxe a seguir para baixar um fluxo de arquivos do Amazon S3 para `stdout`.

Sintaxe

```
$ aws s3 cp <target> [--options] -
```

Para ver algumas opções comuns que podem ser usadas com este comando e exemplos, consulte [Opções usadas com frequência para comandos s3](#). Para obter a lista completa de opções, consulte [s3 cp](#) na Referência de comandos da AWS CLI .

Exemplos do **s3 cp**

O exemplo a seguir copia todos os objetos de `s3://bucket-name/example` para `s3://my-bucket/`.

```
$ aws s3 cp s3://bucket-name/example s3://my-bucket/
```

O exemplo a seguir copia um arquivo local do diretório de trabalho atual para o bucket do Amazon S3 com o comando `s3 cp`.

```
$ aws s3 cp filename.txt s3://bucket-name
```

O exemplo a seguir copia um arquivo do bucket do Amazon S3 para o diretório de trabalho atual, onde `./` especifica o diretório de trabalho atual.

```
$ aws s3 cp s3://bucket-name/filename.txt ./
```

O exemplo a seguir usa `echo` para transmitir o texto "hello world" para o arquivo `s3://bucket-name/filename.txt`.

```
$ echo "hello world" | aws s3 cp - s3://bucket-name/filename.txt
```

O exemplo a seguir transmite o arquivo `s3://bucket-name/filename.txt` para `stdout` e imprime o conteúdo no console.

```
$ aws s3 cp s3://bucket-name/filename.txt -  
hello world
```

O exemplo a seguir transmite o conteúdo de `s3://bucket-name/pre` para `stdout`, usa o comando `bzip2` para comprimir os arquivos e faz upload do novo arquivo compactado chamado `key.bz2` para `s3://bucket-name`.

```
$ aws s3 cp s3://bucket-name/pre - | bzip2 --best | aws s3 cp - s3://bucket-name/key.bz2
```

Sincronizar objetos

O comando [s3 sync](#) sincroniza o conteúdo de um bucket e um diretório ou o conteúdo de dois buckets. Normalmente, `s3 sync` copia arquivos que estão faltando ou desatualizadas ou objetos entre a origem e o destino. No entanto, você também pode fornecer a opção `--delete` para remover arquivos ou objetos do destino que não estão presentes na origem.

Sintaxe

```
$ aws s3 sync <source> <target> [--options]
```

Para ver algumas opções comuns que podem ser usadas com este comando e exemplos, consulte [Opções usadas com frequência para comandos s3](#). Para obter uma lista completa de opções, consulte [s3 sync](#) na Referência de comandos da AWS CLI .

Exemplos de s3 sync

O exemplo a seguir sincroniza o conteúdo de um prefixo do Amazon S3 chamado `path` no bucket chamado `my-bucket` com o diretório de trabalho atual.

`s3 sync` atualiza todos os arquivos que têm tamanho ou hora de modificação diferentes dos arquivos com o mesmo nome no destino. A saída exibe operações específicas executadas durante a sincronização. Observe que a operação sincroniza recursivamente o subdiretório `MySubdirectory` e seu conteúdo com `s3://my-bucket/path/MySubdirectory`.

```
$ aws s3 sync . s3://my-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://my-bucket/path/MySubdirectory/MyFile3.txt
upload: MyFile2.txt to s3://my-bucket/path/MyFile2.txt
upload: MyFile1.txt to s3://my-bucket/path/MyFile1.txt
```

O exemplo a seguir, uma extensão do exemplo anterior, mostra como usar a opção `--delete`.

```
// Delete local file
$ rm ./MyFile1.txt
```

```
// Attempt sync without --delete option - nothing happens
$ aws s3 sync . s3://my-bucket/path

// Sync with deletion - object is deleted from bucket
$ aws s3 sync . s3://my-bucket/path --delete
delete: s3://my-bucket/path/MyFile1.txt

// Delete object from bucket
$ aws s3 rm s3://my-bucket/path/MySubdirectory/MyFile3.txt
delete: s3://my-bucket/path/MySubdirectory/MyFile3.txt

// Sync with deletion - local file is deleted
$ aws s3 sync s3://my-bucket/path . --delete
delete: MySubdirectory\MyFile3.txt

// Sync with Infrequent Access storage class
$ aws s3 sync . s3://my-bucket/path --storage-class STANDARD_IA
```

Quando a opção `--delete` é usada, as opções `--exclude` e `--include` podem filtrar arquivos ou objetos a serem excluídos durante uma operação `s3 sync`. Nesse caso, a sequência de parâmetro deve especificar os arquivos a serem excluídos ou incluídos, a exclusão no contexto do diretório de destino ou bucket. Por exemplo:

```
Assume local directory and s3://my-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://my-bucket/path --delete --exclude "path/MyFile?.txt"
delete: s3://my-bucket/path/MyFile88.txt
...

// Sync with delete, excluding MyFile2.rtf - local file is NOT deleted
$ aws s3 sync s3://my-bucket/path . --delete --exclude "./MyFile2.rtf"
download: s3://my-bucket/path/MyFile1.txt to MyFile1.txt
...

// Sync with delete, local copy of MyFile2.rtf is deleted
$ aws s3 sync s3://my-bucket/path . --delete
```

```
delete: MyFile2.rtf
```

Opções usadas com frequência para comandos s3

As opções a seguir são usadas frequentemente para os comandos descritos neste tópico. Para obter uma lista completa das opções que você pode usar em um comando, consulte o comando específico no guia de [AWS CLI referência da AWS CLI versão 2 do guia](#).

acl

s3 sync e s3 cp podem usar a opção `--acl`. Isso permite definir as permissões de acesso para arquivos copiados para o Amazon S3. A opção `--acl` aceita os valores `private`, `public-read` e `public-read-write`. Para obter mais informações, consulte [Canned ACL](#) no Guia do usuário do Amazon S3.

```
$ aws s3 sync . s3://my-bucket/path --acl public-read
```

exclude

Ao usar o comando s3 cp, s3 mv, s3 sync ou s3 rm, você pode filtrar os resultados usando a opção `--exclude` ou `--include`. A opção `--exclude` define regras para excluir apenas objetos do comando, e as opções se aplicam na ordem especificada. Isso é mostrado no exemplo a seguir.

```
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://my-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://my-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://my-bucket/path --exclude "*.txt" --include "MyFile*.txt" --
exclude "MyFile?.txt"
```


include

Ao usar o comando `s3 cp`, `s3 mv`, `s3 sync` ou `s3 rm`, você pode filtrar os resultados usando a opção `--exclude` ou `--include`. A opção `--include` define regras para incluir apenas objetos especificados para o comando, e as opções se aplicam na ordem especificada. Isso é mostrado no exemplo a seguir.

```
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://my-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://my-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://my-bucket/path --include "*.txt" --exclude "MyFile*.txt" --
include "MyFile?.txt"
```

grant (conceder)

Os comandos `s3 cp`, `s3 mv` e `s3 sync` incluem uma opção `--grants` que pode ser usada para conceder permissões referentes a esse objeto para usuários ou grupos específicos. Defina a opção `--grants` como uma lista de permissões usando a sintaxe a seguir. Substitua `Permission`, `Grantee_Type` e `Grantee_ID` pelos seus próprios valores.

Sintaxe

```
--grants Permission=Grantee_Type=Grantee_ID
         [Permission=Grantee_Type=Grantee_ID ...]
```

Cada valor contém os seguintes elementos:

- *Permission* — Especifica as permissões concedidas. Pode ser definida como `read`, `readacl`, `writeacl` ou `full`.

- *Grantee_Type* — Especifica como identificar o beneficiário. Pode ser definido como `uri`, `emailaddress` ou `id`.
- *Grantee_ID* — Especifica o beneficiário com base em *Grantee_Type*.
 - `uri`— Do grupoURI. Para obter mais informações, consulte [Quem é o favorecido?](#)
 - `emailaddress` - O endereço de e-mail da conta.
 - `id` - O ID canônico da conta

Para obter mais informações sobre controle de acesso do Amazon S3, consulte [Controle de acesso](#).

O exemplo a seguir copia um objeto em um bucket. Ele concede permissões `read` sobre o objeto a todos os usuários e permissões `full` (`read`, `readacl` e `writeacl`) à conta associada a `user@example.com`.

```
$ aws s3 cp file.txt s3://my-bucket/ --grants read=uri=http://acs.amazonaws.com/groups/global/AllUsers full=emailaddress=user@example.com
```

Também será possível especificar uma classe de armazenamento não padrão (`REDUCED_REDUNDANCY` ou `STANDARD_IA`) para objetos cujo upload você fizer no Amazon S3. Para fazer isso, use a opção `--storage-class`.

```
$ aws s3 cp file.txt s3://my-bucket/ --storage-class REDUCED_REDUNDANCY
```

recursive

Quando você usa essa opção, o comando é executado em todos os arquivos ou objetos sob o diretório ou prefixo especificado. O exemplo a seguir exclui `s3://my-bucket/path` e todo o seu conteúdo.

```
$ aws s3 rm s3://my-bucket/path --recursive
```

Recursos

AWS CLI referência:

- [aws s3](#)
- [aws s3 cp](#)

- [aws s3 mb](#)
- [aws s3 mv](#)
- [aws s3 ls](#)
- [aws s3 rb](#)
- [aws s3 rm](#)
- [aws s3 sync](#)

Referência do serviço:

- [Trabalhando com buckets do Amazon S3](#) no Guia do usuário do Amazon S3
- [Trabalhando com objetos do Amazon S3 no Guia](#) do usuário do Amazon S3
- [Listar chaves hierarquicamente usando um prefixo e um delimitador no Guia do usuário do Amazon S3](#)
- [Aborte uploads de várias partes em um bucket do S3 usando o AWS SDK for .NET \(de baixo nível\) no Guia do usuário do Amazon S3](#)

Use os API comandos -Level (s3api) com o AWS CLI

Os comandos API -level (contidos no conjunto de s3api comandos) fornecem acesso direto ao Amazon Simple Storage Service (Amazon S3) APIs e permitem algumas operações que não são expostas nos comandos de alto nível. s3 Esses comandos são equivalentes aos outros AWS serviços que fornecem acesso em API nível superior à funcionalidade dos serviços. Para obter mais informações sobre os comando do s3, consulte [Use comandos de alto nível \(s3\) com o AWS CLI](#)

Este tópico fornece exemplos que demonstram como usar os comandos de nível inferior que são mapeados para o Amazon S3. APIs Além disso, você pode encontrar exemplos para cada API comando do S3 na s3api seção do guia de referência do [guia de AWS CLI referência](#) da .

Tópicos

- [Pré-requisitos](#)
- [Aplicar um personalizado ACL](#)
- [Configurar uma política de registro em log](#)
- [Recursos](#)

Pré-requisitos

Para executar os comandos `s3api`, você precisa:

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI](#) e [Autenticação e credenciais de acesso](#).
- O perfil que você usa deve ter permissões que permitam as AWS operações executadas pelos exemplos.
- Entenda estes termos do Amazon S3:
 - Bucket: uma pasta de nível superior do Amazon S3.
 - Prefixo: uma pasta do Amazon S3 em um bucket.
 - Objeto: qualquer item hospedado em um bucket do Amazon S3.

Aplicar um personalizado ACL

Com comandos de alto nível, você pode usar a `--acl` opção de aplicar listas de controle de acesso predefinidas (ACLs) aos objetos do Amazon S3. Mas você não pode usar esse comando para definir em todo o bucket ACLs. No entanto, você pode fazer isso usando o comando [put-bucket-acl](#) API -level.

O exemplo a seguir mostra como conceder controle total a dois AWS usuários (`user1@example.com` e `user2@example.com`) e permissão de leitura a todos. O identificador para “todos” vem de um especial URI que você passa como parâmetro.

```
$ 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"'
```

Para obter detalhes sobre como construir o ACLs, consulte [PUTBucket acl](#) na API Referência do Amazon Simple Storage Service. Os `s3api` ACL comandos no CLI, como `put-bucket-acl`, usam a mesma notação [abreviada de argumento](#).

Configurar uma política de registro em log

O API comando `put-bucket-logging` configura uma política de registro de buckets.

No exemplo a seguir, o AWS usuário `user@example.com` tem controle total sobre os arquivos de log e todos os usuários têm acesso de leitura a eles. Observe que o `put-bucket-acl` comando

também é necessário para conceder ao sistema de entrega de logs do Amazon S3 (especificado por aURI) as permissões necessárias para ler e gravar os registros no bucket.

```
$ aws s3api put-bucket-acl --bucket MyBucket --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 MyBucket --bucket-logging-status file://logging.json
```

O arquivo `logging.json` no comando anterior tem o seguinte conteúdo.

```
{
  "LoggingEnabled": {
    "TargetBucket": "MyBucket",
    "TargetPrefix": "MyBucketLogs/",
    "TargetGrants": [
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "Type": "Group",
          "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
        },
        "Permission": "READ"
      }
    ]
  }
}
```

Recursos

AWS CLI referência:

- [aws s3api](#)
- [aws s3api put-bucket-acl](#)
- [aws s3api put-bucket-logging](#)

Referência do serviço:

- [Trabalhando com buckets do Amazon S3](#) no Guia do usuário do Amazon S3
- [Trabalhando com objetos do Amazon S3 no Guia](#) do usuário do Amazon S3
- [Listar chaves hierarquicamente usando um prefixo e um delimitador no Guia do usuário do Amazon S3](#)
- [Aborte uploads de várias partes em um bucket do S3 usando o AWS SDK for .NET \(de baixo nível\) no Guia do usuário do Amazon S3](#)

Exemplo de script de operações de ciclo de vida de bucket Amazon S3

Este tópico usa um exemplo de script bash para operações de ciclo de vida do bucket do Amazon S3 usando a AWS Command Line Interface (AWS CLI). Este exemplo de desenvolvimento de scripts usa o conjunto de comandos [aws s3api](#). Scripts shell são programas desenvolvidos para ser executados em uma interface de linha de comando.

Tópicos

- [Antes de começar](#)
- [Sobre este exemplo](#)
- [Arquivos](#)
- [Referências](#)

Antes de começar

Antes que você possa executar qualquer um dos exemplos abaixo, as seguintes tarefas deverão ser concluídas.

- Instale e configure a AWS CLI. Para obter mais informações, consulte [Instalar a AWS CLI](#) e [Autenticação e credenciais de acesso](#).
- O perfil que você usa deve ter permissões que permitam as AWS operações executadas pelos exemplos.
- Como prática AWS recomendada, conceda a esse código o menor privilégio ou somente as permissões necessárias para realizar uma tarefa. Para obter mais informações, consulte [Conceder privilégios mínimos](#) no Guia do IAM usuário.

- Esse código não foi testado em todas as AWS regiões. Alguns AWS serviços estão disponíveis somente em regiões específicas. Para obter mais informações, consulte [Endpoints de serviço e cotas](#) no Guia de referência geral da AWS .
- A execução desse código pode resultar em cobranças em sua AWS conta. É sua responsabilidade garantir que todos os recursos criados por este script sejam removidos após você terminar de usá-los.

O serviço Amazon S3 usa os seguintes termos:

- Bucket: uma pasta de nível superior do Amazon S3.
- Prefixo: uma pasta do Amazon S3 em um bucket.
- Objeto: qualquer item hospedado em um bucket do Amazon S3.

Sobre este exemplo

Este exemplo mostra como interagir com algumas das operações básicas do Amazon S3 usando um conjunto de funções em arquivos de script de shell. As funções estão localizadas no arquivo de script shell chamado `bucket-operations.sh`. Você pode chamar essas funções em outro arquivo. Cada arquivo de script contém comentários descrevendo cada uma das funções.

Para ver os resultados intermediários de cada etapa, execute o script com um parâmetro `-i`. É possível exibir o status atual do bucket ou do conteúdo usando o console do Amazon S3. O script avançará para a próxima etapa somente após você pressionar ENTER no prompt.

Para ver o exemplo completo e os arquivos de script disponíveis para download, consulte [Operações do ciclo de vida do bucket do Amazon S3](#) no repositório de exemplos de código em AWS . GitHub

Arquivos

O exemplo contém os seguintes arquivos:

`bucket-operations.sh`

Esse arquivo de script principal pode ser originado de outro arquivo. Ele inclui funções que executam as seguintes tarefas:

- Criar um bucket e verificar se ele existe
- Copiar um arquivo do computador local para um bucket
- Copiar um arquivo de um local de bucket para um local de bucket diferente

- Listar o conteúdo de um bucket
- Excluir um arquivo de um bucket
- Excluir um bucket

Veja o código para [bucket-operations.sh](#) on GitHub.

test-bucket-operations.sh

O arquivo de script de shell `test-bucket-operations.sh` demonstra como chamar as funções utilizando o arquivo `bucket-operations.sh` e chamando cada uma das funções. Após chamar funções, o script de teste remove todos os recursos que ele criou.

Veja o código para [test-bucket-operations.sh](#) on GitHub.

awsdocs-general.sh

O arquivo de script `awsdocs-general.sh` contém funções de uso geral usadas em exemplos avançados de código para a AWS CLI.

Veja o código para [awsdocs-general.sh](#) on GitHub.

Referências

AWS CLI referência:

- [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](#)

Outra referência:

- [Trabalhando com buckets do Amazon S3](#) no Guia do usuário do Amazon S3

- [Trabalhando com objetos do Amazon S3 no Guia](#) do usuário do Amazon S3
- Para ver, contribuir AWS SDK e AWS CLI codificar exemplos, consulte o [Repositório AWS de exemplos de código](#) em GitHub.

Uso do Amazon SNS com a AWS CLI

É possível acessar os recursos do Amazon Simple Notification Service (Amazon SNS) usando a AWS Command Line Interface (AWS CLI). Para listar os comandos da AWS CLI para o Amazon SNS, use o comando a seguir.

```
aws sns help
```

Antes de executar quaisquer comandos, defina suas credenciais padrão. Para obter mais informações, consulte [Configurar o AWS CLI](#).

Este tópico mostra exemplos de comandos da AWS CLI que executam tarefas comuns para o Amazon SNS.

Tópicos

- [Criar um tópico](#)
- [Assinar um tópico](#)
- [Publicar em um tópico](#)
- [Cancelar a assinatura de um tópico](#)
- [Excluir um tópico](#)

Criar um tópico

Para criar um tópico, use o comando [sns create-topic](#) e especifique o nome a ser atribuído a ele.

```
$ aws sns create-topic --name my-topic
{
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

Anote o TopicArn da resposta que você usará mais tarde para publicar uma mensagem.

Assinar um tópico

Para assinar um tópico, use o comando [sns subscribe](#).

O exemplo a seguir especifica o protocolo `email` e um endereço de e-mail para o `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"
}
```

O AWS envia imediatamente uma mensagem de confirmação para o endereço de e-mail especificado no comando `subscribe`. A mensagem de e-mail tem o seguinte texto.

```
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
```

Assim que o destinatário clicar no link `Confirm subscription` (Confirmar assinatura), o navegador do destinatário exibirá uma mensagem de notificação com informações semelhantes à seguinte.

```
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.
```

Publicar em um tópico

Para enviar uma mensagem a todos os assinantes de um tópico, use o comando [sns publish](#).

O exemplo a seguir envia a mensagem “Hello World!” para todos os assinantes do tópico especificado.

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --  
message "Hello World!"  
{  
  "MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"  
}
```

Neste exemplo, a AWS envia uma mensagem de e-mail com o texto “Hello World!” para `saanvi@example.com`.

Cancelar a assinatura de um tópico

Para cancelar a assinatura de um tópico e parar de receber as mensagens publicadas nesse tópico, use o comando [sns unsubscribe](#) e especifique o ARN do tópico do qual você deseja cancelar a assinatura.

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-  
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

Para verificar se a assinatura foi cancelada com êxito, use o comando [sns list-subscriptions](#) para confirmar que o ARN não aparece mais na lista.

```
$ aws sns list-subscriptions
```

Excluir um tópico

Para excluir um tópico, execute o comando [sns delete-topic](#).

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

Para verificar se a AWS excluiu o tópico com êxito, use o comando [sns list-topics](#) para confirmar que o tópico não aparece mais na lista.

```
$ aws sns list-topics
```

AWS CLI exemplos de comando

Os exemplos de código neste tópico mostram como usar o AWS Command Line Interface with AWS.

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Os cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Serviços

- [ACM exemplos usando AWS CLI](#)
- [API exemplos de gateway usando AWS CLI](#)
- [API Gateway HTTP e WebSocket API exemplos usando AWS CLI](#)
- [API API exemplos de gerenciamento de gateway usando AWS CLI](#)
- [Exemplos de App Mesh usando AWS CLI](#)
- [Exemplos de App Runner usando AWS CLI](#)
- [AWS AppConfig exemplos usando AWS CLI](#)
- [Exemplos de Application Auto Scaling usando AWS CLI](#)
- [Exemplos do Application Discovery Service usando AWS CLI](#)
- [AppRegistry exemplos usando AWS CLI](#)
- [Exemplos de Athena usando AWS CLI](#)
- [Exemplos de Auto Scaling usando AWS CLI](#)
- [Exemplos de planos de Auto Scaling usando AWS CLI](#)
- [AWS Backup exemplos usando AWS CLI](#)
- [AWS Batch exemplos usando AWS CLI](#)
- [AWS Budgets exemplos usando AWS CLI](#)
- [Exemplos do Amazon Chime usando AWS CLI](#)
- [API exemplos do Cloud Control usando AWS CLI](#)
- [AWS Cloud Map exemplos usando AWS CLI](#)
- [AWS Cloud9 exemplos usando AWS CLI](#)
- [AWS CloudFormation exemplos usando AWS CLI](#)

- [CloudFront exemplos usando AWS CLI](#)
- [CloudSearch Exemplos da Amazon usando AWS CLI](#)
- [CloudTrail exemplos usando AWS CLI](#)
- [CloudWatch exemplos usando AWS CLI](#)
- [CloudWatch Exemplos de registros usando AWS CLI](#)
- [CloudWatch Exemplos de monitoramento de rede usando AWS CLI](#)
- [CodeArtifact exemplos usando AWS CLI](#)
- [CodeBuild exemplos usando AWS CLI](#)
- [CodeCommit exemplos usando AWS CLI](#)
- [CodeDeploy exemplos usando AWS CLI](#)
- [CodeGuru Exemplos de revisores usando AWS CLI](#)
- [CodePipeline exemplos usando AWS CLI](#)
- [AWS CodeStar exemplos usando AWS CLI](#)
- [AWS CodeStar Exemplos de notificações usando AWS CLI](#)
- [CodeConnections exemplos usando AWS CLI](#)
- [Exemplos de identidade do Amazon Cognito usando AWS CLI](#)
- [Exemplos de provedores de identidade do Amazon Cognito usando AWS CLI](#)
- [Exemplos do Amazon Comprehend usando AWS CLI](#)
- [Exemplos do Amazon Comprehend Medical usando AWS CLI](#)
- [AWS Config exemplos usando AWS CLI](#)
- [Exemplos do Amazon Connect usando AWS CLI](#)
- [AWS Cost and Usage Report exemplos usando AWS CLI](#)
- [Exemplos do Cost Explorer Service usando AWS CLI](#)
- [Exemplos de Firehose usando AWS CLI](#)
- [Exemplos do Amazon Data Lifecycle Manager usando AWS CLI](#)
- [AWS Data Pipeline exemplos usando AWS CLI](#)
- [DataSync exemplos usando AWS CLI](#)
- [DAXexemplos usando AWS CLI](#)
- [Exemplos de detetives usando AWS CLI](#)

- [Exemplos de Device Farm usando AWS CLI](#)
- [AWS Direct Connect exemplos usando AWS CLI](#)
- [AWS Directory Service exemplos usando AWS CLI](#)
- [AWS DMS exemplos usando AWS CLI](#)
- [Exemplos do Amazon DocumentDB usando AWS CLI](#)
- [Exemplos do DynamoDB usando AWS CLI](#)
- [Exemplos do DynamoDB Streams usando AWS CLI](#)
- [EC2Exemplos da Amazon usando AWS CLI](#)
- [Exemplos do Amazon EC2 Instance Connect usando AWS CLI](#)
- [ECRExemplos da Amazon usando AWS CLI](#)
- [ECSExemplos da Amazon usando AWS CLI](#)
- [EFSExemplos da Amazon usando AWS CLI](#)
- [EKSExemplos da Amazon usando AWS CLI](#)
- [Exemplos do Elastic Beanstalk usando AWS CLI](#)
- [Elastic Load Balancing — Exemplos da versão 1 usando AWS CLI](#)
- [Elastic Load Balancing - Exemplos da versão 2 usando AWS CLI](#)
- [Exemplos do Elastic Transcoder usando AWS CLI](#)
- [ElastiCache exemplos usando AWS CLI](#)
- [MediaStore exemplos usando AWS CLI](#)
- [EMRExemplos da Amazon usando AWS CLI](#)
- [Amazon EMR em EKS exemplos usando AWS CLI](#)
- [EventBridge exemplos usando AWS CLI](#)
- [Exemplos do Firewall Manager usando AWS CLI](#)
- [AWS FIS exemplos usando AWS CLI](#)
- [GameLift Exemplos da Amazon usando AWS CLI](#)
- [Exemplos do Global Accelerator usando AWS CLI](#)
- [AWS Glue exemplos usando AWS CLI](#)
- [GuardDuty exemplos usando AWS CLI](#)
- [AWS Health exemplos usando AWS CLI](#)

- [HealthImaging exemplos usando AWS CLI](#)
- [HealthLake exemplos usando AWS CLI](#)
- [HealthOmics exemplos usando AWS CLI](#)
- [IAMexemplos usando AWS CLI](#)
- [IAMExemplos do Access Analyzer usando AWS CLI](#)
- [Exemplos do Image Builder usando AWS CLI](#)
- [Exemplos do Incident Manager usando AWS CLI](#)
- [Exemplos de contatos do Incident Manager usando AWS CLI](#)
- [Exemplos do Amazon Inspector usando AWS CLI](#)
- [AWS IoT exemplos usando AWS CLI](#)
- [AWS IoT 1-Click Exemplos de dispositivos usando AWS CLI](#)
- [AWS IoT 1-Click Exemplos de projetos usando AWS CLI](#)
- [AWS IoT Analytics exemplos usando AWS CLI](#)
- [Exemplos do Device Advisor usando AWS CLI](#)
- [AWS IoT data exemplos usando AWS CLI](#)
- [AWS IoT Events exemplos usando AWS CLI](#)
- [AWS IoT Events-Data exemplos usando AWS CLI](#)
- [AWS IoT Greengrass exemplos usando AWS CLI](#)
- [AWS IoT Greengrass V2 exemplos usando AWS CLI](#)
- [AWS IoT Jobs SDK release exemplos usando AWS CLI](#)
- [AWS IoT SiteWise exemplos usando AWS CLI](#)
- [AWS IoT Things Graph exemplos usando AWS CLI](#)
- [AWS IoT Wireless exemplos usando AWS CLI](#)
- [IVSExemplos da Amazon usando AWS CLI](#)
- [Exemplos do Amazon IVS Chat usando AWS CLI](#)
- [Exemplos de streaming IVS em tempo real da Amazon usando AWS CLI](#)
- [Exemplos do Amazon Kendra usando AWS CLI](#)
- [Exemplos de Kinesis usando AWS CLI](#)
- [AWS KMS exemplos usando AWS CLI](#)

- [Exemplos de Lake Formation usando AWS CLI](#)
- [Exemplos de Lambda usando AWS CLI](#)
- [Exemplos de License Manager usando AWS CLI](#)
- [Exemplos do Lightsail usando AWS CLI](#)
- [Exemplos de Macie usando AWS CLI](#)
- [Exemplos de Amazon Managed Grafana usando AWS CLI](#)
- [MediaConnect exemplos usando AWS CLI](#)
- [MediaConvert exemplos usando AWS CLI](#)
- [MediaLive exemplos usando AWS CLI](#)
- [MediaPackage exemplos usando AWS CLI](#)
- [MediaPackage VOD exemplos usando AWS CLI](#)
- [MediaStore Exemplos de planos de dados usando AWS CLI](#)
- [MediaTailor exemplos usando AWS CLI](#)
- [Exemplos de MemoryDB usando AWS CLI](#)
- [MSK Exemplos da Amazon usando AWS CLI](#)
- [Exemplos do Network Manager usando AWS CLI](#)
- [Exemplos do Nimble Studio usando AWS CLI](#)
- [OpenSearch Exemplos de serviços usando AWS CLI](#)
- [AWS OpsWorks exemplos usando AWS CLI](#)
- [AWS OpsWorks CM exemplos usando AWS CLI](#)
- [Exemplos de organizações usando AWS CLI](#)
- [AWS Outposts exemplos usando AWS CLI](#)
- [AWS Payment Cryptography exemplos usando AWS CLI](#)
- [AWS Payment Cryptography Exemplos de planos de dados usando AWS CLI](#)
- [Exemplos do Amazon Pinpoint usando AWS CLI](#)
- [Exemplos do Amazon Polly usando AWS CLI](#)
- [AWS Price List exemplos usando AWS CLI](#)
- [AWS Private CA exemplos usando AWS CLI](#)
- [AWS Proton exemplos usando AWS CLI](#)

- [QLDBexemplos usando AWS CLI](#)
- [RDSExemplos da Amazon usando AWS CLI](#)
- [Exemplos do Amazon RDS Data Service usando AWS CLI](#)
- [Exemplos do Amazon RDS Performance Insights usando AWS CLI](#)
- [Exemplos do Amazon Redshift usando AWS CLI](#)
- [Exemplos do Amazon Rekognition usando AWS CLI](#)
- [AWS RAM exemplos usando AWS CLI](#)
- [Exemplos do Resource Explorer usando AWS CLI](#)
- [Exemplos de Resource Groups usando AWS CLI](#)
- [APIExemplos de marcação de Resource Groups usando AWS CLI](#)
- [AWS RoboMaker exemplos usando AWS CLI](#)
- [Exemplos do Route 53 usando AWS CLI](#)
- [Exemplos de registro de domínio do Route 53 usando AWS CLI](#)
- [Exemplos de resolvedores do Route 53 usando AWS CLI](#)
- [Exemplos do Amazon S3 usando AWS CLI](#)
- [Exemplos de controle do Amazon S3 usando AWS CLI](#)
- [Exemplos do S3 Glacier usando AWS CLI](#)
- [Exemplos de Secrets Manager usando AWS CLI](#)
- [Exemplos do Security Hub usando AWS CLI](#)
- [AWS Serverless Application Repository exemplos usando AWS CLI](#)
- [Exemplos do Service Catalog usando AWS CLI](#)
- [Exemplos de Service Quotas usando AWS CLI](#)
- [SESExemplos da Amazon usando AWS CLI](#)
- [Exemplos de Shield usando AWS CLI](#)
- [Exemplos de signatários usando AWS CLI](#)
- [Exemplos de Snowball usando AWS CLI](#)
- [SNSExemplos da Amazon usando AWS CLI](#)
- [SQSExemplos da Amazon usando AWS CLI](#)
- [Exemplos de Storage Gateway usando AWS CLI](#)

- [AWS STS exemplos usando AWS CLI](#)
- [AWS Support exemplos usando AWS CLI](#)
- [SWFExemplos da Amazon usando AWS CLI](#)
- [Exemplos do Systems Manager usando AWS CLI](#)
- [Exemplos do Amazon Textract usando AWS CLI](#)
- [Exemplos do Amazon Transcribe usando AWS CLI](#)
- [Exemplos do Amazon Translate usando AWS CLI](#)
- [Trusted Advisor exemplos usando AWS CLI](#)
- [Exemplos de permissões verificadas usando AWS CLI](#)
- [VPCExemplos de treliça usando AWS CLI](#)
- [AWS WAF Classic exemplos usando AWS CLI](#)
- [AWS WAF Classic regional exemplos usando AWS CLI](#)
- [AWS WAFV2 exemplos usando AWS CLI](#)
- [WorkDocs Exemplos da Amazon usando AWS CLI](#)
- [WorkMail Exemplos da Amazon usando AWS CLI](#)
- [Exemplos de fluxo de WorkMail mensagens da Amazon usando AWS CLI](#)
- [WorkSpaces exemplos usando AWS CLI](#)
- [Exemplos de raio-X usando AWS CLI](#)

ACM exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with ACM.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-certificate

O código de exemplo a seguir mostra como usar `add-tags-to-certificate`.

AWS CLI

Para adicionar tags a um ACM certificado existente

O comando `add-tags-to-certificate` a seguir adiciona duas tags ao certificado especificado. Use um espaço para separar várias tags:

```
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
```

- Para API obter detalhes, consulte [AddTagsToCertificate](#) na Referência de AWS CLI Comandos.

delete-certificate

O código de exemplo a seguir mostra como usar `delete-certificate`.

AWS CLI

Para excluir um ACM certificado da sua conta

O `delete-certificate` comando a seguir exclui o certificado com o especificado ARN:

```
aws acm delete-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- Para API obter detalhes, consulte [DeleteCertificate](#) na Referência de AWS CLI Comandos.

describe-certificate

O código de exemplo a seguir mostra como usar `describe-certificate`.

AWS CLI

Para recuperar os campos contidos em um certificado ACM

O `describe-certificate` comando a seguir recupera todos os campos do certificado com o especificadoARN:

```
aws acm describe-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Uma saída semelhante à seguinte é exibida:

```
{
  "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"
    ]
  }
}

```

- Para API obter detalhes, consulte [DescribeCertificate](#) na Referência de AWS CLI Comandos.

export-certificate

O código de exemplo a seguir mostra como usar `export-certificate`.

AWS CLI

Para exportar um certificado privado emitido por uma CA privada.

O `export-certificate` comando a seguir exporta um certificado privado, uma cadeia de certificados e uma chave privada para sua tela:

```

aws acm export-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file

```

Para exportar o certificado, a cadeia e a chave privada para um arquivo local, use o seguinte 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

```

- Para API obter detalhes, consulte [ExportCertificate](#) na Referência de AWS CLI Comandos.

get-certificate

O código de exemplo a seguir mostra como usar `get-certificate`.

AWS CLI

Para recuperar um certificado ACM

O `get-certificate` comando a seguir recupera o certificado do especificado ARN e da cadeia de certificados:

```
aws acm get-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Uma saída semelhante à seguinte é exibida:

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
```


import-certificate

O código de exemplo a seguir mostra como usar `import-certificate`.

AWS CLI

Para importar um certificado para ACM o.

O `import-certificate` comando a seguir importa um certificado para ACM o. Substitua os nomes dos arquivos pelos seus próprios:

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- Para API obter detalhes, consulte [ImportCertificate](#) na Referência de AWS CLI Comandos.

list-certificates

O código de exemplo a seguir mostra como usar `list-certificates`.

AWS CLI

Para listar os ACM certificados de uma AWS conta

O `list-certificates` comando a seguir ARNs lista os certificados em sua conta:

```
aws acm list-certificates
```

O comando anterior produz uma saída semelhante à seguinte:

```
{
  "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"
    }
  ]
}
```



```

    }
  ]
}

```

Você pode decidir quantos certificados deseja exibir sempre que chamar `list-certificates`. Por exemplo, se você tem quatro certificados e deseja exibir apenas dois por vez, defina o argumento `max-items` como 2, conforme o seguinte exemplo:

```
aws acm list-certificates --max-items 2
```

Dois certificados ARNs e um NextToken valor serão exibidos:

```

"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"

```

Para exibir os próximos dois certificados em sua conta, defina o valor NextToken na próxima chamada:

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-  
b58e-2b837bd9ba48
```

É possível filtrar a saída usando o argumento `certificate-statuses`. O comando a seguir exibe certificados com VALIDATION status PENDING _:

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

Também é possível filtrar a saída usando o argumento `includes`. O comando a seguir exibe os certificados filtrados nas propriedades a seguir. Os certificados a serem exibidos:

- 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
```

- Para API obter detalhes, consulte [ListCertificates](#) na Referência de AWS CLI Comandos.

list-tags-for-certificate

O código de exemplo a seguir mostra como usar `list-tags-for-certificate`.

AWS CLI

Para listar as tags aplicadas a um ACM certificado

O seguinte comando `list-tags-for-certificate` lista as tags aplicadas a um certificado na conta:

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

O comando anterior produz uma saída semelhante à seguinte:

```
{
  "Tags": [
    {
      "Value": "Website",
      "Key": "Purpose"
    },
    {
      "Value": "Alice",
      "Key": "Admin"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListTagsForCertificate](#) na Referência de AWS CLI Comandos.

remove-tags-from-certificate

O código de exemplo a seguir mostra como usar `remove-tags-from-certificate`.

AWS CLI

Para remover uma tag de um ACM certificado

O comando `remove-tags-from-certificate` a seguir remove duas tags do certificado especificado. Use um espaço para separar várias tags:

```
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
```

- Para API obter detalhes, consulte [RemoveTagsFromCertificate](#) na Referência de AWS CLI Comandos.

request-certificate

O código de exemplo a seguir mostra como usar `request-certificate`.

AWS CLI

Para solicitar um novo ACM certificado

O `request-certificate` comando a seguir solicita um novo certificado para o domínio `www.example.com` usando a validação: DNS

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

Você pode inserir um token de idempotência para distinguir entre chamadas para `request-certificate`:

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --idempotency-token 91adc45q
```

Você pode inserir um ou mais nomes alternativos de assunto para solicitar um certificado que protegerá mais de um domínio do apex:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.net
```

Você pode inserir um nome alternativo que também pode ser usado para acessar seu site:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

Você pode usar um asterisco (*) como um caractere curinga para criar um certificado para vários subdomínios no mesmo domínio:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

Você também pode inserir vários nomes alternativos:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
subject-alternative-names b.example.com c.example.com d.example.com
```

Se você estiver usando e-mail para validação, poderá inserir as opções de validação de domínio para especificar o domínio para o qual o e-mail de validação será enviado:

```
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
```

O seguinte comando desativa o registro em log de transparência do certificado ao solicitar um novo certificado:

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --  
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- Para API obter detalhes, consulte [RequestCertificate](#) na Referência de AWS CLI Comandos.

resend-validation-email

O código de exemplo a seguir mostra como usar `resend-validation-email`.

AWS CLI

Para reenviar o e-mail de validação para sua solicitação ACM de certificado

O seguinte `resend-validation-email` comando instrui a autoridade de certificação da Amazon a enviar e-mails de validação para os endereços apropriados:

```
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
```

- Para API obter detalhes, consulte [ResendValidationEmail](#) na Referência de AWS CLI Comandos.

update-certificate-options

O código de exemplo a seguir mostra como usar `update-certificate-options`.

AWS CLI

Para atualizar as opções de certificado

O `update-certificate-options` comando a seguir exclui o registro de transparência do certificado:

```
aws acm update-certificate-options --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --options CertificateTransparencyLoggingPreference=DISABLED
```

- Para API obter detalhes, consulte [UpdateCertificateOptions](#) na Referência de AWS CLI Comandos.

API Exemplos de gateway usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with API Gateway.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-api-key

O código de exemplo a seguir mostra como usar `create-api-key`.

AWS CLI

Para criar uma API chave que esteja habilitada para um Palco existente API

Comando:

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for
development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- Para API obter detalhes, consulte [CreateApiKey](#) na Referência de AWS CLI Comandos.

create-authorizer

O código de exemplo a seguir mostra como usar `create-authorizer`.

AWS CLI

Exemplo 1: Para criar um API Gateway Custom Authorizer baseado em token para o API

O `create-authorizer` exemplo a seguir cria um autorizador baseado em 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
```

Saída:

```
{
  "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"
}
```

Exemplo 2: Para criar um Autorizador Personalizado de API Gateway baseado em Grupos de Usuários do Cognito para o API

O `create-authorizer` exemplo a seguir cria um Autorizador Personalizado de API Gateway baseado em Grupos de Usuários do Cognito.

```
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'
```

Saída:

```
{
  "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"
}
```

Exemplo 3: Para criar um API Gateway Custom Authorizer baseado em solicitações para o API

O `create-authorizer` exemplo a seguir cria um autorizador baseado em solicitações.

```
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
```

Saída:

```
{
  "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
}
```

- Para API obter detalhes, consulte [CreateAuthorizer](#) na Referência de AWS CLI Comandos.

create-base-path-mapping

O código de exemplo a seguir mostra como usar `create-base-path-mapping`.

AWS CLI

Para criar o mapeamento do caminho base para um nome de domínio personalizado

Comando:

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-
api-id 1234123412 --stage prod --base-path v1
```

- Para API obter detalhes, consulte [CreateBasePathMapping](#) na Referência de AWS CLI Comandos.

create-deployment

O código de exemplo a seguir mostra como usar `create-deployment`.

AWS CLI

Para implantar os recursos configurados API para um novo estágio

Comando:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-description 'Development Stage' --description 'First deployment to the dev stage'
```

Para implantar os recursos configurados API para um estágio existente

Comando:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Second deployment to the dev stage'
```

Para implantar os recursos configurados API para um estágio existente com variáveis de estágio

```
aws apigateway create-deployment -- rest-api-id 1234123412 --stage-name dev --description 'Terceira implantação no estágio de desenvolvimento' --variables key='value', ='otherKeyotherValue'
```

- Para API obter detalhes, consulte [CreateDeployment](#) na Referência de AWS CLI Comandos.

create-domain-name

O código de exemplo a seguir mostra como usar `create-domain-name`.

AWS CLI

Para criar o nome de domínio personalizado

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'
```

- Para API obter detalhes, consulte [CreateDomainName](#) na Referência de AWS CLI Comandos.

create-model

O código de exemplo a seguir mostra como usar create-model.

AWS CLI

Para criar um modelo para um 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" } } } } }'
```

Saída:

```
{
  "contentType": "application/json",
  "description": "The First Model",
  "name": "firstModel",
  "id": "2rzg01",
  "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty\": { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string\" } } } } }"
```

- Para API obter detalhes, consulte [CreateModel](#) na Referência de AWS CLI Comandos.

create-resource

O código de exemplo a seguir mostra como usar create-resource.

AWS CLI

Para criar um recurso em um API

Comando:

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-part 'new-resource'
```

- Para API obter detalhes, consulte [CreateResource](#) na Referência de AWS CLI Comandos.

create-rest-api

O código de exemplo a seguir mostra como usar create-rest-api.

AWS CLI

Para criar um API

Comando:

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first API'
```

Para criar uma duplicata a API partir de uma existente 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
```

- Para API obter detalhes, consulte [CreateRestApi](#) na Referência de AWS CLI Comandos.

create-stage

O código de exemplo a seguir mostra como usar create-stage.

AWS CLI

Para criar um estágio em um API que conterà uma implantação existente

Comando:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --description 'Development stage' --deployment-id a1b2c3
```

Para criar um estágio em um API que conterá uma implantação existente e variáveis de estágio personalizadas

Comando:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'  
--description 'Development stage' --deployment-id a1b2c3 --variables  
key='value',otherKey='otherValue'
```

- Para API obter detalhes, consulte [CreateStage](#) na Referência de AWS CLI Comandos.

create-usage-plan-key

O código de exemplo a seguir mostra como usar create-usage-plan-key.

AWS CLI

Associar uma API chave existente a um plano de uso

Comando:

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --  
key-id 4vq3yryqm5
```

- Para API obter detalhes, consulte [CreateUsagePlanKey](#) na Referência de AWS CLI Comandos.

create-usage-plan

O código de exemplo a seguir mostra como usar create-usage-plan.

AWS CLI

Para criar um plano de uso com limites de aceleração e cota que são redefinidos no início do mês

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
```

- Para API obter detalhes, consulte [CreateUsagePlan](#) na Referência de AWS CLI Comandos.

delete-api-key

O código de exemplo a seguir mostra como usar `delete-api-key`.

AWS CLI

Para excluir uma API chave

Comando:

```
aws apigateway delete-api-key --api-key 8bk1k8b1k3sB38D9B310enyWT8c09B301kq0b1k
```

- Para API obter detalhes, consulte [DeleteApiKey](#) na Referência de AWS CLI Comandos.

delete-authorizer

O código de exemplo a seguir mostra como usar `delete-authorizer`.

AWS CLI

Para excluir um Autorizador Personalizado em um API

Comando:

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- Para API obter detalhes, consulte [DeleteAuthorizer](#) na Referência de AWS CLI Comandos.

delete-base-path-mapping

O código de exemplo a seguir mostra como usar `delete-base-path-mapping`.

AWS CLI

Para excluir um mapeamento de caminho base para um nome de domínio personalizado

Comando:

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path 'dev'
```

- Para API obter detalhes, consulte [DeleteBasePathMapping](#) na Referência de AWS CLI Comandos.

delete-client-certificate

O código de exemplo a seguir mostra como usar `delete-client-certificate`.

AWS CLI

Para excluir um certificado de cliente

Comando:

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- Para API obter detalhes, consulte [DeleteClientCertificate](#) na Referência de AWS CLI Comandos.

delete-deployment

O código de exemplo a seguir mostra como usar `delete-deployment`.

AWS CLI

Para excluir uma implantação em um API

Comando:

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- Para API obter detalhes, consulte [DeleteDeployment](#) na Referência de AWS CLI Comandos.

delete-domain-name

O código de exemplo a seguir mostra como usar `delete-domain-name`.

AWS CLI

Para excluir um nome de domínio personalizado

Comando:

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- Para API obter detalhes, consulte [DeleteDomainName](#) na Referência de AWS CLI Comandos.

delete-integration-response

O código de exemplo a seguir mostra como usar `delete-integration-response`.

AWS CLI

Para excluir uma resposta de integração para um determinado recurso, método e código de status em um API

Comando:

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200
```

- Para API obter detalhes, consulte [DeleteIntegrationResponse](#) na Referência de AWS CLI Comandos.

delete-integration

O código de exemplo a seguir mostra como usar `delete-integration`.

AWS CLI

Para excluir uma integração para um determinado recurso e método em um API

Comando:

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- Para API obter detalhes, consulte [DeleteIntegration](#) na Referência de AWS CLI Comandos.

delete-method-response

O código de exemplo a seguir mostra como usar `delete-method-response`.

AWS CLI

Para excluir uma resposta de método para o determinado recurso, método e código de status em um API

Comando:

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3
--http-method GET --status-code 200
```

- Para API obter detalhes, consulte [DeleteMethodResponse](#) na Referência de AWS CLI Comandos.

delete-method

O código de exemplo a seguir mostra como usar `delete-method`.

AWS CLI

Para excluir um método para um determinado recurso em um API

Comando:

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET
```

- Para API obter detalhes, consulte [DeleteMethod](#) na Referência de AWS CLI Comandos.

delete-model

O código de exemplo a seguir mostra como usar `delete-model`.

AWS CLI

Para excluir um modelo em determinado API

Comando:

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- Para API obter detalhes, consulte [DeleteModel](#) na Referência de AWS CLI Comandos.

delete-resource

O código de exemplo a seguir mostra como usar `delete-resource`.

AWS CLI

Para excluir um recurso em um API

Comando:

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- Para API obter detalhes, consulte [DeleteResource](#) na Referência de AWS CLI Comandos.

delete-rest-api

O código de exemplo a seguir mostra como usar `delete-rest-api`.

AWS CLI

Para excluir um API

Comando:

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- Para API obter detalhes, consulte [DeleteRestApi](#) na Referência de AWS CLI Comandos.

delete-stage

O código de exemplo a seguir mostra como usar `delete-stage`.

AWS CLI

Para excluir um estágio em um API

Comando:

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- Para API obter detalhes, consulte [DeleteStage](#) na Referência de AWS CLI Comandos.

delete-usage-plan-key

O código de exemplo a seguir mostra como usar delete-usage-plan-key.

AWS CLI

Para remover uma API chave de um plano de uso

Comando:

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- Para API obter detalhes, consulte [DeleteUsagePlanKey](#) na Referência de AWS CLI Comandos.

delete-usage-plan

O código de exemplo a seguir mostra como usar delete-usage-plan.

AWS CLI

Para excluir um plano de uso

Comando:

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- Para API obter detalhes, consulte [DeleteUsagePlan](#) na Referência de AWS CLI Comandos.

flush-stage-authorizers-cache

O código de exemplo a seguir mostra como usar flush-stage-authorizers-cache.

AWS CLI

Para limpar todas as entradas do cache do autorizador em um palco

Comando:

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-name dev
```

- Para API obter detalhes, consulte [FlushStageAuthorizersCache](#) na Referência de AWS CLI Comandos.

flush-stage-cache

O código de exemplo a seguir mostra como usar `flush-stage-cache`.

AWS CLI

Para limpar o cache do estágio API de um

Comando:

```
aws apigateway flush-stage-cache --rest-api-id 1234123412 --stage-name dev
```

- Para API obter detalhes, consulte [FlushStageCache](#) na Referência de AWS CLI Comandos.

generate-client-certificate

O código de exemplo a seguir mostra como usar `generate-client-certificate`.

AWS CLI

Para criar um certificado do lado do cliente SSL

Comando:

```
aws apigateway generate-client-certificate --description 'My First Client Certificate'
```

- Para API obter detalhes, consulte [GenerateClientCertificate](#) na Referência de AWS CLI Comandos.

get-account

O código de exemplo a seguir mostra como usar `get-account`.

AWS CLI

Para obter as configurações da conta do API Gateway

Comando:

```
aws apigateway get-account
```

Saída:

```
{
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
  APIGatewayToCloudWatchLogsRole",
  "throttleSettings": {
    "rateLimit": 500.0,
    "burstLimit": 1000
  }
}
```

- Para API obter detalhes, consulte [GetAccount](#) na Referência de AWS CLI Comandos.

get-api-key

O código de exemplo a seguir mostra como usar `get-api-key`.

AWS CLI

Para obter as informações sobre uma API chave específica

Comando:

```
aws apigateway get-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k
```

Saída:

```
{
  "description": "My first key",
  "enabled": true,
  "stageKeys": [
    "a1b2c3d4e5/dev",
    "e5d4c3b2a1/dev"
  ],
  "lastUpdatedDate": 1456184515,
  "createdDate": 1456184452,
  "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k",
  "name": "My key"
}
```

- Para API obter detalhes, consulte [GetApiKey](#) na Referência de AWS CLI Comandos.

get-api-keys

O código de exemplo a seguir mostra como usar `get-api-keys`.

AWS CLI

Para obter a lista de API chaves

Comando:

```
aws apigateway get-api-keys
```

Saída:

```
{
  "items": [
    {
      "description": "My first key",
      "enabled": true,
      "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
      ],
      "lastUpdatedDate": 1456184515,
      "createdDate": 1456184452,
      "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k",
      "name": "My key"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetApiKeys](#) na Referência de AWS CLI Comandos.

get-authorizer

O código de exemplo a seguir mostra como usar `get-authorizer`.

AWS CLI

Para obter as configurações do API Gateway per- API Authorizer

Comando:

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetAuthorizer](#) na Referência de AWS CLI Comandos.

get-authorizers

O código de exemplo a seguir mostra como usar `get-authorizers`.

AWS CLI

Para obter a lista de autorizadores de um REST API

Comando:

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

Saída:

```
{
  "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"  
      }  
    ]  
  }
```

- Para API obter detalhes, consulte [GetAuthorizers](#) na Referência de AWS CLI Comandos.

get-base-path-mapping

O código de exemplo a seguir mostra como usar `get-base-path-mapping`.

AWS CLI

Como obter o mapeamento de caminho base para um nome de domínio personalizado

Comando:

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-path v1
```

Saída:

```
{  
  "basePath": "v1",  
  "restApiId": "1234w4321e",  
  "stage": "api"  
}
```

- Para API obter detalhes, consulte [GetBasePathMapping](#) na Referência de AWS CLI Comandos.

get-base-path-mappings

O código de exemplo a seguir mostra como usar `get-base-path-mappings`.

AWS CLI

Para obter os mapeamentos de caminho base para um nome de domínio personalizado

Comando:

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

Saída:

```
{
  "items": [
    {
      "basePath": "(none)",
      "restApiId": "1234w4321e",
      "stage": "dev"
    },
    {
      "basePath": "v1",
      "restApiId": "1234w4321e",
      "stage": "api"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetBasePathMappings](#) na Referência de AWS CLI Comandos.

get-client-certificate

O código de exemplo a seguir mostra como usar `get-client-certificate`.

AWS CLI

Para obter um certificado de cliente

Comando:

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- Para API obter detalhes, consulte [GetClientCertificate](#) na Referência de AWS CLI Comandos.

get-client-certificates

O código de exemplo a seguir mostra como usar `get-client-certificates`.

AWS CLI

Para obter uma lista de certificados de clientes

Comando:

```
aws apigateway get-client-certificates
```

Saída:

```
{
  "items": [
    {
      "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
content> -----END CERTIFICATE-----",
      "clientCertificateId": "a1b2c3",
      "expirationDate": 1483556561,
      "description": "My Client Certificate",
      "createdDate": 1452020561
    }
  ]
}
```

- Para API obter detalhes, consulte [GetClientCertificates](#) na Referência de AWS CLI Comandos.

get-deployment

O código de exemplo a seguir mostra como usar `get-deployment`.

AWS CLI

Para obter informações sobre uma implantação

Comando:

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

Saída:

```
{
  "description": "myDeployment",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

- Para API obter detalhes, consulte [GetDeployment](#) na Referência de AWS CLI Comandos.

get-deployments

O código de exemplo a seguir mostra como usar `get-deployments`.

AWS CLI

Para obter uma lista de implantações para um REST API

Comando:

```
aws apigateway get-deployments --rest-api-id 1234123412
```

Saída:

```
{
  "items": [
    {
      "createdDate": 1453797217,
      "id": "0a2b4c",
      "description": "Deployed my API for the first time"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetDeployments](#) na Referência de AWS CLI Comandos.

get-domain-name

O código de exemplo a seguir mostra como usar `get-domain-name`.

AWS CLI

Para obter informações sobre um nome de domínio personalizado

Comando:

```
aws apigateway get-domain-name --domain-name api.domain.tld
```

Saída:

```
{
  "domainName": "api.domain.tld",
}
```

```
"distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
"certificateName": "uploadedCertificate",
"certificateUploadDate": 1462565487
}
```

- Para API obter detalhes, consulte [GetDomainName](#) na Referência de AWS CLI Comandos.

get-domain-names

O código de exemplo a seguir mostra como usar `get-domain-names`.

AWS CLI

Para obter uma lista de nomes de domínio personalizados

Comando:

```
aws apigateway get-domain-names
```

Saída:

```
{
  "items": [
    {
      "distributionDomainName": "d9511k3109bkd.cloudfront.net",
      "certificateUploadDate": 1452812505,
      "certificateName": "my_custom_domain-certificate",
      "domainName": "subdomain.domain.tld"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetDomainNames](#) na Referência de AWS CLI Comandos.

get-export

O código de exemplo a seguir mostra como usar `get-export`.

AWS CLI

Para obter o modelo JSON Swagger para um palco

Comando:

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Para obter o modelo JSON Swagger + API Gateway Extensions para um estágio

Comando:

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Para obter o modelo JSON Swagger + Postman Extensions para um palco

Comando:

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

- Para API obter detalhes, consulte [GetExport](#) na Referência de AWS CLI Comandos.

get-integration-response

O código de exemplo a seguir mostra como usar `get-integration-response`.

AWS CLI

Para obter a configuração da resposta de integração para um HTTP método definido no recurso REST API de a

Comando:

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

Saída:

```
{
  "statusCode": "200",
  "responseTemplates": {
```

```
    "application/json": null
  }
}
```

- Para API obter detalhes, consulte [GetIntegrationResponse](#) na Referência de AWS CLI Comandos.

get-integration

O código de exemplo a seguir mostra como usar `get-integration`.

AWS CLI

Para obter a configuração de integração para um HTTP método definido no recurso REST API de a

Comando:

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetIntegration](#) na Referência de AWS CLI Comandos.

get-method-response

O código de exemplo a seguir mostra como usar `get-method-response`.

AWS CLI

Para obter a configuração do recurso de resposta do método para um HTTP método definido no recurso REST API de a

Comando:

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

Saída:

```
{
  "responseModels": {
    "application/json": "Empty"
  },
  "statusCode": "200"
}
```

- Para API obter detalhes, consulte [GetMethodResponse](#) na Referência de AWS CLI Comandos.

get-method

O código de exemplo a seguir mostra como usar `get-method`.

AWS CLI

Para obter a configuração do recurso de método para um HTTP método definido no recurso REST API de a

Comando:

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetMethod](#) na Referência de AWS CLI Comandos.

get-model-template

O código de exemplo a seguir mostra como usar `get-model-template`.

AWS CLI

Para obter o modelo de mapeamento para um modelo definido em REST API

Comando:

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

Saída:

```
{
  "value": "#set($inputRoot = $input.path('$'))\n{ }"
```

- Para API obter detalhes, consulte [GetModelTemplate](#) na Referência de AWS CLI Comandos.

get-model

O código de exemplo a seguir mostra como usar `get-model`.

AWS CLI

Para obter a configuração de um modelo definido em REST API

Comando:

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

Saída:

```
{
  "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}"
```

- Para API obter detalhes, consulte [GetModel](#) na Referência de AWS CLI Comandos.

get-models

O código de exemplo a seguir mostra como usar `get-models`.

AWS CLI

Para obter uma lista de modelos para um REST API

Comando:

```
aws apigateway get-models --rest-api-id 1234123412
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetModels](#) na Referência de AWS CLI Comandos.

get-resource

O código de exemplo a seguir mostra como usar `get-resource`.

AWS CLI

Para obter informações sobre um recurso

Comando:

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

Saída:

```
{
  "path": "/path",
  "pathPart": "path",
  "id": "zwo0y3",
  "parentId": "uyokt6ij2g"
}
```

- Para API obter detalhes, consulte [GetResource](#) na Referência de AWS CLI Comandos.

get-resources

O código de exemplo a seguir mostra como usar `get-resources`.

AWS CLI

Para obter uma lista de recursos para um REST API

Comando:

```
aws apigateway get-resources --rest-api-id 1234123412
```

Saída:

```
{
  "items": [
    {
      "path": "/resource/subresource",
      "resourceMethods": {
        "POST": {}
      },
      "id": "024ace",
      "pathPart": "subresource",
      "parentId": "ai5b02"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetResources](#) na Referência de AWS CLI Comandos.

get-rest-api

O código de exemplo a seguir mostra como usar `get-rest-api`.

AWS CLI

Para obter informações sobre um API

Comando:

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

Saída:

```
{
  "name": "myAPI",
  "id": "o1y243m4f5",
  "createdDate": 1453416433
}
```

- Para API obter detalhes, consulte [GetRestApi](#) na Referência de AWS CLI Comandos.

get-rest-apis

O código de exemplo a seguir mostra como usar `get-rest-apis`.

AWS CLI

Para obter uma lista de REST APIs

Comando:

```
aws apigateway get-rest-apis
```

Saída:

```
{
  "items": [
    {
```

```

        "createdDate": 1438884790,
        "id": "12s44z21rb",
        "name": "My First API"
    }
]
}

```

- Para API obter detalhes, consulte [GetRestApis](#) na Referência de AWS CLI Comandos.

get-sdk

O código de exemplo a seguir mostra como usar `get-sdk`.

AWS CLI

Para obter o Android SDK para um REST API estágio

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

```

Saída:

```

{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}

```

Para obter o IOS SDK para um 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

```

Saída:

```

{

```

```
"contentType": "application/octet-stream",
"contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip\"
}"
```

Para obter o Javascript SDK para um REST API estágio

Comando:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type javascript /path/to/javascript_sdk.zip
```

Saída:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip\"
}"
```

- Para API obter detalhes, consulte [GetSdk](#) na Referência de AWS CLI Comandos.

get-stage

O código de exemplo a seguir mostra como usar `get-stage`.

AWS CLI

Para obter informações sobre o estágio API de um

Comando:

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

Saída:

```
{
  "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
      }
    }
  }
}

```

- Para API obter detalhes, consulte [GetStage](#) na Referência de AWS CLI Comandos.

get-stages

O código de exemplo a seguir mostra como usar get-stages.

AWS CLI

Para obter a lista de estágios de um REST API

Comando:

```
aws apigateway get-stages --rest-api-id 1234123412
```

Saída:

```
{
  "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
        }
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [GetStages](#) na Referência de AWS CLI Comandos.

get-usage-plan-key

O código de exemplo a seguir mostra como usar `get-usage-plan-key`.

AWS CLI

Para obter os detalhes de uma API chave associada a um plano de uso

Comando:

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- Para API obter detalhes, consulte [GetUsagePlanKey](#) na Referência de AWS CLI Comandos.

get-usage-plan-keys

O código de exemplo a seguir mostra como usar `get-usage-plan-keys`.

AWS CLI

Para obter a lista de API chaves associadas a um plano de uso

Comando:

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- Para API obter detalhes, consulte [GetUsagePlanKeys](#) na Referência de AWS CLI Comandos.

get-usage-plan

O código de exemplo a seguir mostra como usar `get-usage-plan`.

AWS CLI

Para obter os detalhes de um plano de uso

Comando:

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- Para API obter detalhes, consulte [GetUsagePlan](#) na Referência de AWS CLI Comandos.

get-usage-plans

O código de exemplo a seguir mostra como usar `get-usage-plans`.

AWS CLI

Para obter os detalhes de todos os planos de uso

Comando:

```
aws apigateway get-usage-plans
```

- Para API obter detalhes, consulte [GetUsagePlans](#) na Referência de AWS CLI Comandos.

get-usage

O código de exemplo a seguir mostra como usar `get-usage`.

AWS CLI

Para obter os detalhes de uso de um plano de uso

Comando:

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-date "2016-08-17"
```

- Para API obter detalhes, consulte [GetUsage](#) na Referência de AWS CLI Comandos.

import-rest-api

O código de exemplo a seguir mostra como usar `import-rest-api`.

AWS CLI

Para importar um modelo do Swagger e criar um API

Comando:

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- Para API obter detalhes, consulte [ImportRestApi](#) na Referência de AWS CLI Comandos.

put-integration-response

O código de exemplo a seguir mostra como usar `put-integration-response`.

AWS CLI

Como criar uma resposta de integração como a resposta padrão com um modelo de mapeamento definido

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\"}"}'
```

Como criar uma resposta de integração com um regex de 400 e um valor de cabeçalho definido de forma estática

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": ""}'
```

- Para API obter detalhes, consulte [PutIntegrationResponse](#) na Referência de AWS CLI Comandos.

put-integration

O código de exemplo a seguir mostra como usar put-integration.

AWS CLI

Para criar uma solicitação MOCK de integração

Comando:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type MOCK --request-templates '{"application/json": "{\"statusCode\": 200}"}'
```

Para criar uma solicitação HTTP de integração

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'
```

Para criar uma solicitação de AWS integração com um endpoint da Função 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'
```

- Para API obter detalhes, consulte [PutIntegration](#) na Referência de AWS CLI Comandos.

put-method-response

O código de exemplo a seguir mostra como usar `put-method-response`.

AWS CLI

Como criar uma resposta de método de acordo com o código de status especificado com um cabeçalho de resposta de método personalizado

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"
```

- Para API obter detalhes, consulte [PutMethodResponse](#) na Referência de AWS CLI Comandos.

put-method

O código de exemplo a seguir mostra como usar `put-method`.

AWS CLI

Para criar um método para um recurso em um API cabeçalho de solicitação de método personalizado sem autorização, sem API chave

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"
```

- Para API obter detalhes, consulte [PutMethod](#) na Referência de AWS CLI Comandos.

put-rest-api

O código de exemplo a seguir mostra como usar `put-rest-api`.

AWS CLI

Para sobrescrever um existente API usando um modelo Swagger

Comando:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body
'fileb:///path/to/API_Swagger_template.json'
```

Para mesclar um modelo do Swagger em um existente API

Comando:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///
path/to/API_Swagger_template.json'
```

- Para API obter detalhes, consulte [PutRestApi](#) na Referência de AWS CLI Comandos.

test-invoke-authorizer

O código de exemplo a seguir mostra como usar `test-invoke-authorizer`.

AWS CLI

Para testar, invoque uma solicitação para um Autorizador Personalizado, incluindo o cabeçalho e o valor necessários.

Comando:

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-
id 5yid1t --headers Authorization='Value'
```

- Para API obter detalhes, consulte [TestInvokeAuthorizer](#) na Referência de AWS CLI Comandos.

test-invoke-method

O código de exemplo a seguir mostra como usar `test-invoke-method`.

AWS CLI

Para testar, invoque o recurso raiz em um API fazendo uma solicitação GET

Comando:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id av15sg8fw8
--http-method GET --path-with-query-string '/'
```

Para testar, invocar um sub-recurso em um API fazendo uma GET solicitação com um valor de parâmetro de caminho especificado

Comando:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- Para API obter detalhes, consulte [TestInvokeMethod](#) na Referência de AWS CLI Comandos.

update-account

O código de exemplo a seguir mostra como usar `update-account`.

AWS CLI

Para alterar a IAM função ARN de registro em CloudWatch registros

Comando:

```
aws apigateway update-account --patch-operations op='replace',path='/
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

Saída:

```
{
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
  APIGatewayToCloudWatchLogs",
  "throttleSettings": {
    "rateLimit": 1000.0,
    "burstLimit": 2000
  }
}
```

- Para API obter detalhes, consulte [UpdateAccount](#) na Referência de AWS CLI Comandos.

update-api-key

O código de exemplo a seguir mostra como usar `update-api-key`.

AWS CLI

Para alterar o nome de uma API chave

Comando:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/name',value='newName'
```

Saída:

```
{
  "description": "currentDescription",
  "enabled": true,
  "stageKeys": [
    "41t2j324r5/dev"
  ],
  "lastUpdatedDate": 1470086052,
  "createdDate": 1445460347,
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
  "name": "newName"
}
```

Para desativar a API chave

Comando:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --  
patch-operations op='replace',path='/enabled',value='false'
```

Saída:

```
{  
  "description": "currentDescription",  
  "enabled": false,  
  "stageKeys": [  
    "41t2j324r5/dev"  
  ],  
  "lastUpdatedDate": 1470086052,  
  "createdDate": 1445460347,  
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",  
  "name": "newName"  
}
```

- Para API obter detalhes, consulte [UpdateApiKey](#) na Referência de AWS CLI Comandos.

update-authorizer

O código de exemplo a seguir mostra como usar update-authorizer.

AWS CLI

Para alterar o nome do Autorizador Personalizado

Comando:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --  
patch-operations op='replace',path='/name',value='testAuthorizer'
```

Saída:

```
{  
  "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"
}
```

Para alterar a Função Lambda que é invocada pelo Autorizador Personalizado

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'
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [UpdateAuthorizer](#) na Referência de AWS CLI Comandos.

update-base-path-mapping

O código de exemplo a seguir mostra como usar update-base-path-mapping.

AWS CLI

Como alterar o caminho base de um nome de domínio personalizado

Comando:

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

Saída:


```
{
  "basePath": "v1",
  "restApiId": "1234123412",
  "stage": "api"
}
```

- Para API obter detalhes, consulte [UpdateBasePathMapping](#) na Referência de AWS CLI Comandos.

update-client-certificate

O código de exemplo a seguir mostra como usar `update-client-certificate`.

AWS CLI

Para atualizar a descrição de um certificado de cliente

Comando:

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-operations op='replace',path='/description',value='My new description'
```

- Para API obter detalhes, consulte [UpdateClientCertificate](#) na Referência de AWS CLI Comandos.

update-deployment

O código de exemplo a seguir mostra como usar `update-deployment`.

AWS CLI

Para alterar a descrição de uma implantação

Comando:

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --patch-operations op='replace',path='/description',value='newDescription'
```

Saída:

```
{
```

```

    "description": "newDescription",
    "id": "ztt4m2",
    "createdDate": 1455218022
  }

```

- Para API obter detalhes, consulte [UpdateDeployment](#) na Referência de AWS CLI Comandos.

update-domain-name

O código de exemplo a seguir mostra como usar `update-domain-name`.

AWS CLI

Para alterar o nome do certificado para um nome de domínio personalizado

O `update-domain-name` exemplo a seguir altera o nome do certificado para um domínio personalizado.

```

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'

```

Saída:

```

{
  "domainName": "api.domain.tld",
  "distributionDomainName": "d123456789012.cloudfront.net",
  "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/
CERTEXAMPLE123EXAMPLE",
  "certificateUploadDate": 1462565487
}

```

Para obter mais informações, consulte [Configurar um nome de domínio personalizado para um API gateway API no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [UpdateDomainName](#) na Referência de AWS CLI Comandos.

update-integration-response

O código de exemplo a seguir mostra como usar `update-integration-response`.

AWS CLI

Para alterar um cabeçalho de resposta de integração para ter um mapeamento estático de '*'

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='''''*''''
```

Saída:

```
{  
  "statusCode": "200",  
  "responseParameters": {  
    "method.response.header.Access-Control-Allow-Origin": "''*''"  
  }  
}
```

Para remover um cabeçalho de resposta de integração

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'
```

- Para API obter detalhes, consulte [UpdateIntegrationResponse](#) na Referência de AWS CLI Comandos.

update-integration

O código de exemplo a seguir mostra como usar update-integration.

AWS CLI

Para adicionar o modelo de mapeamento 'Content-Type: application/json' configurado com passagem de entrada

Comando:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='add',path='/requestTemplates/application~1json'"
```

Para atualizar (substituir) o modelo de mapeamento 'Content-Type: application/json' configurado por um modelo personalizado

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\"}'"
```

Para atualizar (substituir) um modelo personalizado associado a 'Content-Type: application/json' com 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'"
```

Para remover o modelo de mapeamento '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'"
```

- Para API obter detalhes, consulte [UpdateIntegration](#) na Referência de AWS CLI Comandos.

update-method-response

O código de exemplo a seguir mostra como usar `update-method-response`.

AWS CLI

Para criar um novo cabeçalho de resposta de método para a resposta 200 em um método e defini-lo como não obrigatório (padrão)

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"
```

Para excluir um modelo de resposta para a resposta 200 em um método

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"
```

- Para API obter detalhes, consulte [UpdateMethodResponse](#) na Referência de AWS CLI Comandos.

update-method

O código de exemplo a seguir mostra como usar `update-method`.

AWS CLI

Exemplo 1: Para modificar um método para exigir uma API chave

O `update-method` exemplo a seguir modifica o método para exigir uma API chave.

```
aws apigateway update-method \  
  --rest-api-id 1234123412 \  
  --resource-id a1b2c3 \  
  --http-method GET \  
  --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

Saída:

```
{
  "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": {}
      }
    }
  }
}
```

Exemplo 2: Para modificar um método para exigir IAM autorização

O `update-method` exemplo a seguir modifica o método para exigir IAM autorização.

```
aws apigateway update-method \
  --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"
```

Saída:

```
{
  "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": {}
      }
    }
  }
}
```

Exemplo 3: Modificar um método para exigir autorização Lambda

O `update-method` exemplo a seguir modifica o método para a autorização Lambda necessária.

```
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"
```

Saída:

```
{
  "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": {}
        }
      }
    }
  }
}

```

Para obter mais informações, consulte [Criar, configurar e testar planos de uso usando o API Gateway CLI e Controlar REST API e gerenciar o acesso a um API gateway REST API no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [UpdateMethod](#) na Referência de AWS CLI Comandos.

update-model

O código de exemplo a seguir mostra como usar `update-model`.

AWS CLI

Para alterar a descrição de um modelo em um API

Comando:


```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/description,value='New Description'
```

Para alterar o esquema de um modelo em um 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\" }''
```

- Para API obter detalhes, consulte [UpdateModel](#) na Referência de AWS CLI Comandos.

update-resource

O código de exemplo a seguir mostra como usar `update-resource`.

AWS CLI

Para mover um recurso e colocá-lo sob um recurso principal diferente em um API

Comando:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/parentId,value='3c2b1a'
```

Saída:

```
{
  "path": "/resource",
  "pathPart": "resource",
  "id": "1a2b3c",
  "parentId": "3c2b1a"
}
```

Para renomear um recurso (`pathPart`) em um API

Comando:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/pathPart,value=newresourceName
```

Saída:

```
{
  "path": "/newresource",
  "pathPart": "newresource",
  "id": "1a2b3c",
  "parentId": "3c2b1a"
}
```

- Para API obter detalhes, consulte [UpdateResource](#) na Referência de AWS CLI Comandos.

update-rest-api

O código de exemplo a seguir mostra como usar `update-rest-api`.

AWS CLI

Para alterar o nome de um API

Comando:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
op=replace,path=/name,value='New Name'
```

Para alterar a descrição de um API

Comando:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
op=replace,path=/description,value='New Description'
```

- Para API obter detalhes, consulte [UpdateRestApi](#) na Referência de AWS CLI Comandos.

update-stage

O código de exemplo a seguir mostra como usar `update-stage`.

AWS CLI

Exemplo 1: Para substituir as configurações de estágio de um recurso e método

O `update-stage` exemplo a seguir substitui as configurações do estágio e desativa o registro completo de solicitação/resposta para um recurso e método específicos.

```
aws apigateway update-stage \  
  --rest-api-id 1234123412 \  
  --stage-name 'dev' \  
  --patch-operations op=replace,path=~1resourceName/GET/logging/  
dataTrace,value=false
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Como configurar um palco para um REST API](#) no Guia do desenvolvedor do Amazon API Gateway.

Exemplo 2: Para atualizar as configurações do estágio para todos os recursos e métodos de um API estágio

O `update-stage` exemplo a seguir ativa o registro completo de solicitação/resposta para todos os recursos e métodos de um estágio. API

```
aws apigateway update-stage \  
  --rest-api-id 1234123412 \  
  --stage-name 'dev' \  
  --patch-operations 'op=replace,path=/**/logging/dataTrace,value=true'
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Como configurar um palco para um REST API](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [UpdateStage](#) na Referência de AWS CLI Comandos.

update-usage-plan

O código de exemplo a seguir mostra como usar `update-usage-plan`.

AWS CLI

Para alterar o período definido em um Plano de Uso

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/period",value="MONTH"
```

Para alterar o limite de cota definido em um plano de uso

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/limit",value="500"
```

Para alterar o limite da taxa de aceleração definido em um Plano de Uso

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/rateLimit",value="10"
```

Para alterar o limite de pico de aceleração definido em um Plano de Uso

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/burstLimit",value="20"
```

- Para API obter detalhes, consulte [UpdateUsagePlan](#) na Referência de AWS CLI Comandos.

update-usage

O código de exemplo a seguir mostra como usar update-usage.

AWS CLI

Para modificar temporariamente a cota em uma API chave para o período atual definido no Plano de Uso

Comando:

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/remaining",value="50"
```

- Para API obter detalhes, consulte [UpdateUsage](#) na Referência de AWS CLI Comandos.

APIGateway HTTP e WebSocket API exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with API Gateway HTTP WebSocket API e.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-api-mapping

O código de exemplo a seguir mostra como usar `create-api-mapping`.

AWS CLI

Para criar um API mapeamento para um API

O `create-api-mapping` exemplo a seguir mapeia o `test` estágio de an API para o `/myApi` caminho do nome de domínio `regional.example.com` personalizado.

```
aws apigatewayv2 create-api-mapping \  
  --domain-name regional.example.com \  
  --api-mapping-key myApi \  
  --api-id a1b2c3d4 \  
  --stage test
```

Saída:

```
{
  "ApiId": "a1b2c3d4",
  "ApiMappingId": "0qzs2sy7bh",
  "ApiMappingKey": "myApi"
  "Stage": "test"
}
```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [CreateApiMapping](#) na Referência de AWS CLI Comandos.

create-api

O código de exemplo a seguir mostra como usar `create-api`.

AWS CLI

Para criar um HTTP API

O `create-api` exemplo a seguir cria um HTTP API usando a criação rápida. Você pode usar a criação rápida para criar uma API com um AWS Lambda ou HTTP integração, uma rota abrangente padrão e um estágio padrão configurado para implantar alterações automaticamente. O comando a seguir usa a criação rápida para criar uma HTTP API que se integre a uma função Lambda.

```
aws apigatewayv2 create-api \
  --name my-http-api \
  --protocol-type HTTP \
  --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

Saída:

```
{
  "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"  
}
```

Para obter mais informações, consulte [Desenvolvendo um HTTP API no API Gateway](#) no Guia do desenvolvedor do Amazon API Gateway.

Para criar um WebSocket API

O `create-api` exemplo a seguir cria um WebSocket API com o nome especificado.

```
aws apigatewayv2 create-api \  
  --name "myWebSocketApi" \  
  --protocol-type WEBSOCKET \  
  --route-selection-expression '$request.body.action'
```

Saída:

```
{  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "Name": "myWebSocketApi",  
  "CreateDate": "2018-11-15T06:23:51Z",  
  "ProtocolType": "WEBSOCKET",  
  "RouteSelectionExpression": "'$request.body.action'",  
  "ApiId": "aabbccdde"  
}
```

Para obter mais informações, consulte [Create a WebSocket API in API Gateway](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [CreateApi](#) na Referência de AWS CLI Comandos.

create-authorizer

O código de exemplo a seguir mostra como usar `create-authorizer`.

AWS CLI

Para criar um JWT autorizador para um HTTP API

O `create-authorizer` exemplo a seguir cria um JWT autorizador que usa o Amazon Cognito como provedor de identidade.


```
aws apigatewayv2 create-authorizer \  
  --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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Controle do acesso HTTP APIs com JWT autorizadores](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [CreateAuthorizer](#) na Referência de AWS CLI Comandos.

create-deployment

O código de exemplo a seguir mostra como usar create-deployment.

AWS CLI

Para criar uma implantação para um API

O create-deployment exemplo a seguir cria uma implantação para um API e associa essa implantação ao dev estágio do API.

```
aws apigatewayv2 create-deployment \  
  --api-id a1b2c3d4 \  
  --stage-name dev \  
  --deployment-name my-deployment
```

```
--api-id a1b2c3d4 \  
--stage-name dev
```

Saída:

```
{  
  "AutoDeployed": false,  
  "CreatedDate": "2020-04-06T23:38:08Z",  
  "DeploymentId": "531z91",  
  "DeploymentStatus": "DEPLOYED"  
}
```

Para obter mais informações, consulte a [APIimplantação](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [CreateDeployment](#) na Referência de AWS CLI Comandos.

create-domain-name

O código de exemplo a seguir mostra como usar create-domain-name.

AWS CLI

Para criar um nome de domínio personalizado

O create-domain-name exemplo a seguir cria um nome de domínio personalizado regional para umAPI.

```
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
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [CreateDomainName](#) na Referência de AWS CLI Comandos.

create-integration

O código de exemplo a seguir mostra como usar `create-integration`.

AWS CLI

Para criar uma WebSocket API integração

O `create-integration` exemplo a seguir cria uma integração simulada para um WebSocket API.

```

aws apigatewayv2 create-integration \
  --api-id aabbccdee \
  --passthrough-behavior WHEN_NO_MATCH \
  --timeout-in-millis 29000 \
  --connection-type INTERNET \
  --integration-type MOCK

```

Saída:

```

{
  "ConnectionType": "INTERNET",
  "IntegrationId": "0abcdef",
  "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
  "IntegrationType": "MOCK",
  "PassthroughBehavior": "WHEN_NO_MATCH",
  "PayloadFormatVersion": "1.0",

```

```
"TimeoutInMillis": 29000
}
```

Para obter mais informações, consulte [Configurar uma solicitação de WebSocket API integração no API Gateway no](#) Guia do desenvolvedor do Amazon API Gateway.

Para criar uma HTTP API integração

O `create-integration` exemplo a seguir cria uma integração AWS Lambda para um. 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
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Como configurar integrações HTTP APIs](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [CreateIntegration](#) na Referência de AWS CLI Comandos.

create-route

O código de exemplo a seguir mostra como usar `create-route`.

AWS CLI

Para criar uma rota `$default` para um ou WebSocket HTTP API

O `create-route` exemplo a seguir cria uma `$default` rota para um WebSocket ou HTTPAPI.

```
aws apigatewayv2 create-route \  
  --api-id aabbccdee \  
  --route-key '$default'
```

Saída:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteKey": "$default",  
  "RouteId": "1122334"  
}
```

Para obter mais informações, consulte Como [trabalhar com rotas WebSocket APIs](#) no Guia do desenvolvedor do Amazon API Gateway

Para criar uma rota para um HTTP API

O `create-route` exemplo a seguir cria uma rota chamada `signup` que aceita POST solicitações.

```
aws apigatewayv2 create-route \  
  --api-id aabbccdee \  
  --route-key 'POST /signup'
```

Saída:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteKey": "POST /signup",  
  "RouteId": "1122334"  
}
```

Para obter mais informações, consulte Como [trabalhar com rotas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway

- Para API obter detalhes, consulte [CreateRoute](#) na Referência de AWS CLI Comandos.

create-stage

O código de exemplo a seguir mostra como usar `create-stage`.

AWS CLI

Para criar um palco

O `create-stage` exemplo a seguir cria um estágio chamado `dev` para um API.

```
aws apigatewayv2 create-stage \  
  --api-id a1b2c3d4 \  
  --stage-name dev
```

Saída:

```
{  
  "CreateDate": "2020-04-06T23:23:46Z",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "LastUpdatedDate": "2020-04-06T23:23:46Z",  
  "RouteSettings": {},  
  "StageName": "dev",  
  "StageVariables": {},  
  "Tags": {}  
}
```

Para obter mais informações, consulte Como [trabalhar com estágios HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [CreateStage](#) na Referência de AWS CLI Comandos.

create-vpc-link

O código de exemplo a seguir mostra como usar `create-vpc-link`.

AWS CLI

Para criar um VPC link para um HTTP API

O `create-vpc-link` exemplo a seguir cria um VPC link para HTTP APIs.

```
aws apigatewayv2 create-vpc-link \  
  --name MyVpcLink \  
  --subnet-ids subnet-aaaa subnet-bbbb \  
  --security-group-ids sg1234 sg5678
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte Como [trabalhar com VPC links HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [CreateVpcLink](#) na Referência de AWS CLI Comandos.

delete-access-log-settings

O código de exemplo a seguir mostra como usar `delete-access-log-settings`.

AWS CLI

Para desativar o registro de acesso para um API

O `delete-access-log-settings` exemplo a seguir exclui as configurações do registro de acesso para o `$default` estágio de um API. Para desativar o registro de acesso de um estágio, exclua suas configurações de registro de acesso.

```
aws apigatewayv2 delete-access-log-settings \  
  --api-id a1b2c3d4 \  
  --stage-name '$default'
```

Este comando não produz saída.

Para obter mais informações, consulte [Como configurar o registro em log para um HTTP API](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [DeleteAccessLogSettings](#) na Referência de AWS CLI Comandos.

delete-api-mapping

O código de exemplo a seguir mostra como usar `delete-api-mapping`.

AWS CLI

Para excluir um API mapeamento

O `delete-api-mapping` exemplo a seguir exclui um API mapeamento para o nome de domínio `api.example.com` personalizado.

```
aws apigatewayv2 delete-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [DeleteApiMapping](#) na Referência de AWS CLI Comandos.

delete-api

O código de exemplo a seguir mostra como usar `delete-api`.

AWS CLI

Para excluir um API

O `delete-api` exemplo a seguir exclui um API.

```
aws apigatewayv2 delete-api \  
  --api-id a1b2c3d4
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com HTTP APIs](#) e [trabalhando com WebSocket APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [DeleteApi](#) na Referência de AWS CLI Comandos.

delete-authorizer

O código de exemplo a seguir mostra como usar `delete-authorizer`.

AWS CLI

Para excluir um autorizador

O `delete-authorizer` exemplo a seguir exclui um autorizador.

```
aws apigatewayv2 delete-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3
```

Este comando não produz saída.

Para obter mais informações, consulte [Controle do acesso HTTP APIs com JWT autorizadores](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [DeleteAuthorizer](#) na Referência de AWS CLI Comandos.

delete-cors-configuration

O código de exemplo a seguir mostra como usar `delete-cors-configuration`.

AWS CLI

Para excluir a CORS configuração de um HTTP API

O `delete-cors-configuration` exemplo a seguir CORS desabilita um HTTP API excluindo sua CORS configuração.

```
aws apigatewayv2 delete-cors-configuration \  
  --api-id a1b2c3d4
```

Este comando não produz saída.

Para obter mais informações, consulte [Como configurar CORS em HTTP API](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [DeleteCorsConfiguration](#) na Referência de AWS CLI Comandos.

delete-deployment

O código de exemplo a seguir mostra como usar delete-deployment.

AWS CLI

Para excluir uma implantação

O delete-deployment exemplo a seguir exclui a implantação de um API.

```
aws apigatewayv2 delete-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id a1b2c3
```

Este comando não produz saída.

Para obter mais informações, consulte a [API Implantação](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [DeleteDeployment](#) na Referência de AWS CLI Comandos.

delete-domain-name

O código de exemplo a seguir mostra como usar delete-domain-name.

AWS CLI

Para excluir um nome de domínio personalizado

O delete-domain-name exemplo a seguir exclui um nome de domínio personalizado.

```
aws apigatewayv2 delete-domain-name \  
  --domain-name api.example.com
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [DeleteDomainName](#) na Referência de AWS CLI Comandos.

delete-integration

O código de exemplo a seguir mostra como usar delete-integration.

AWS CLI

Para excluir uma integração

O delete-integration exemplo a seguir exclui uma API integração.

```
aws apigatewayv2 delete-integration \  
  --api-id a1b2c3d4 \  
  --integration-id a1b2c3
```

Este comando não produz saída.

Para obter mais informações, consulte [Como configurar integrações HTTP APIs e Configurar integrações no Amazon API Gateway WebSocket API Developer Guide](#).

- Para API obter detalhes, consulte [DeleteIntegration](#) na Referência de AWS CLI Comandos.

delete-route-settings

O código de exemplo a seguir mostra como usar delete-route-settings.

AWS CLI

Para excluir as configurações de rota

O delete-route-settings exemplo a seguir exclui as configurações de rota para a rota especificada.

```
aws apigatewayv2 delete-route-settings \  
  --api-id a1b2c3d4 \  
  --stage-name dev \  
  --route-key 'GET /pets'
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com rotas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [DeleteRouteSettings](#) na Referência de AWS CLI Comandos.

delete-route

O código de exemplo a seguir mostra como usar delete-route.

AWS CLI

Para excluir uma rota

O delete-route exemplo a seguir exclui uma API rota.

```
aws apigatewayv2 delete-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com rotas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [DeleteRoute](#) na Referência de AWS CLI Comandos.

delete-stage

O código de exemplo a seguir mostra como usar delete-stage.

AWS CLI

Para excluir um estágio

O delete-stage exemplo a seguir exclui o test estágio de um API.

```
aws apigatewayv2 delete-stage \  
  --api-id a1b2c3d4 \  
  --stage-name test
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com estágios HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [DeleteStage](#) na Referência de AWS CLI Comandos.

delete-vpc-link

O código de exemplo a seguir mostra como usar `delete-vpc-link`.

AWS CLI

Para excluir um VPC link para um HTTP API

O `delete-vpc-link` exemplo a seguir exclui um VPC link.

```
aws apigatewayv2 delete-vpc-link \  
  --vpc-link-id abcd123
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com VPC links HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [DeleteVpcLink](#) na Referência de AWS CLI Comandos.

export-api

O código de exemplo a seguir mostra como usar `export-api`.

AWS CLI

Para exportar uma API definição aberta de um HTTP API

O `export-api` exemplo a seguir exporta uma definição do Open API 3.0 de um API estágio chamado `prod` para um YAML arquivo chamado `stage-definition.yaml`. O arquivo de definição exportado inclui extensões do API Gateway por padrão.

```
aws apigatewayv2 export-api \  
  --api-id a1b2c3d4 \  
  --output-type YAML \  
  --specification OAS30 \  
  --stage-name prod \  
  stage-definition.yaml
```

Este comando não produz saída.

Para obter mais informações, consulte [Exportando um HTTP API do API Gateway](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [ExportApi](#) na Referência de AWS CLI Comandos.

get-api-mapping

O código de exemplo a seguir mostra como usar `get-api-mapping`.

AWS CLI

Para obter informações sobre um API mapeamento para um nome de domínio personalizado

O `get-api-mapping` exemplo a seguir exibe informações sobre um API mapeamento para o nome de domínio `api.example.com` personalizado.

```
aws apigatewayv2 get-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

Saída:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "a1b2c3d5",  
  "ApiMappingKey": "myTestApi"  
  "Stage": "test"  
}
```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [GetApiMapping](#) na Referência de AWS CLI Comandos.

get-api-mappings

O código de exemplo a seguir mostra como usar `get-api-mappings`.

AWS CLI

Para obter API mapeamentos para um nome de domínio personalizado

O `get-api-mappings` exemplo a seguir exibe uma lista de todos os API mapeamentos para o nome de domínio `api.example.com` personalizado.

```
aws apigatewayv2 get-api-mappings \  
  --domain-name api.example.com
```

Saída:

```
{  
  "Items": [  
    {  
      "ApiId": "a1b2c3d4",  
      "ApiMappingId": "a1b2c3d5",  
      "ApiMappingKey": "myTestApi"  
      "Stage": "test"  
    },  
    {  
      "ApiId": "a5b6c7d8",  
      "ApiMappingId": "a1b2c3d6",  
      "ApiMappingKey": "myDevApi"  
      "Stage": "dev"  
    },  
  ],  
}
```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [GetApiMappings](#) na Referência de AWS CLI Comandos.

get-api

O código de exemplo a seguir mostra como usar `get-api`.

AWS CLI

Para recuperar informações sobre um API

O `get-api` exemplo a seguir exibe informações sobre um API.

```
aws apigatewayv2 get-api \  
  --api-id a1b2c3d4
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [GetApi](#) na Referência de AWS CLI Comandos.

get-apis

O código de exemplo a seguir mostra como usar `get-apis`.

AWS CLI

Para recuperar uma lista de APIs

O `get-apis` exemplo a seguir lista todos APIs os do usuário atual.

```
aws apigatewayv2 get-apis
```

Saída:

```
{  
  "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": {}
    }
  ]
}

```

Para obter mais informações, consulte [Trabalhando com HTTP APIs](#) e [trabalhando com WebSocket APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [GetApis](#) na Referência de AWS CLI Comandos.

get-authorizer

O código de exemplo a seguir mostra como usar `get-authorizer`.

AWS CLI

Para recuperar informações sobre um autorizador

O `get-authorizer` exemplo a seguir exibe informações sobre um autorizador.

```

aws apigatewayv2 get-authorizer \
  --api-id a1b2c3d4 \
  --authorizer-id a1b2c3

```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Controle do acesso HTTP APIs com JWT autorizadores](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [GetAuthorizer](#) na Referência de AWS CLI Comandos.

get-authorizers

O código de exemplo a seguir mostra como usar `get-authorizers`.

AWS CLI

Para recuperar uma lista de autorizadores para um API

O `get-authorizers` exemplo a seguir exibe uma lista de todos os autorizadores de um API.

```
aws apigatewayv2 get-authorizers \
  --api-id a1b2c3d4
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Controle do acesso HTTP APIs com JWT autorizadores](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [GetAuthorizers](#) na Referência de AWS CLI Comandos.

get-deployment

O código de exemplo a seguir mostra como usar get-deployment.

AWS CLI

Para recuperar informações sobre uma implantação

O get-deployment exemplo a seguir exibe informações sobre uma implantação.

```
aws apigatewayv2 get-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id abcdef
```

Saída:

```
{  
  "AutoDeployed": true,  
  "CreatedDate": "2020-04-07T23:58:40Z",  
  "DeploymentId": "abcdef",  
  "DeploymentStatus": "DEPLOYED",  
  "Description": "Automatic deployment triggered by changes to the Api  
  configuration"  
}
```

Para obter mais informações, consulte a [APIimplantação](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [GetDeployment](#) na Referência de AWS CLI Comandos.

get-deployments

O código de exemplo a seguir mostra como usar get-deployments.

AWS CLI

Para recuperar uma lista de implantações

O get-deployments exemplo a seguir exibe uma lista de todas as implantações API de um.

```
aws apigatewayv2 get-deployments \  
  --api-id a1b2c3d4
```

Saída:

```
{  
  "Items": [  
    {  
      "AutoDeployed": true,  
      "CreatedDate": "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"
    }
]
}

```

Para obter mais informações, consulte a [APIimplantação](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [GetDeployments](#) na Referência de AWS CLI Comandos.

get-domain-name

O código de exemplo a seguir mostra como usar `get-domain-name`.

AWS CLI

Para recuperar informações sobre um nome de domínio personalizado

O `get-domain-name` exemplo a seguir exibe informações sobre um nome de domínio personalizado.

```
aws apigatewayv2 get-domain-name \
  --domain-name api.example.com
```

Saída:

```

{
  "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": {}
}

```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [GetDomainName](#) na Referência de AWS CLI Comandos.

get-domain-names

O código de exemplo a seguir mostra como usar `get-domain-names`.

AWS CLI

Para recuperar uma lista de nomes de domínio personalizados

O `get-domain-names` exemplo a seguir exibe uma lista de todos os nomes de domínio personalizados do usuário atual.

```
aws apigatewayv2 get-domain-names
```

Saída:

```

{
  "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"
        }
    ]
}
]
}
}

```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [GetDomainNames](#) na Referência de AWS CLI Comandos.

get-integration

O código de exemplo a seguir mostra como usar `get-integration`.

AWS CLI

Para recuperar informações sobre uma integração

O `get-integration` exemplo a seguir exibe informações sobre uma integração.

```
aws apigatewayv2 get-integration \
  --api-id a1b2c3d4 \
```

```
--integration-id a1b2c3
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Como configurar integrações HTTP APIs e Configurar integrações no Amazon API Gateway WebSocket API Developer Guide](#).

- Para API obter detalhes, consulte [GetIntegration](#) na Referência de AWS CLI Comandos.

get-integrations

O código de exemplo a seguir mostra como usar `get-integrations`.

AWS CLI

Para recuperar uma lista de integrações

O `get-integrations` exemplo a seguir exibe uma lista de todas as integrações API de um.

```
aws apigatewayv2 get-integrations \
  --api-id a1b2c3d4
```

Saída:

```
{
  "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
  }
]
```

Para obter mais informações, consulte [Como configurar integrações HTTP APIs e Configurar integrações no Amazon API Gateway WebSocket API Developer Guide](#).

- Para API obter detalhes, consulte [GetIntegrations](#) na Referência de AWS CLI Comandos.

get-route

O código de exemplo a seguir mostra como usar `get-route`.

AWS CLI

Para recuperar informações sobre uma rota

O `get-route` exemplo a seguir exibe informações sobre uma rota.

```
aws apigatewayv2 get-route \
  --api-id a1b2c3d4 \
  --route-id 72jz1wk
```

Saída:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
```

```
"RouteId": "72jz1wk",  
"RouteKey": "ANY /pets",  
"Target": "integrations/a1b2c3"  
}
```

Para obter mais informações, consulte Como [trabalhar com rotas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway

- Para API obter detalhes, consulte [GetRoute](#) na Referência de AWS CLI Comandos.

get-routes

O código de exemplo a seguir mostra como usar `get-routes`.

AWS CLI

Para recuperar uma lista de rotas

O `get-routes` exemplo a seguir exibe uma lista de todas as rotas API de um.

```
aws apigatewayv2 get-routes \  
  --api-id a1b2c3d4
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
}
```

Para obter mais informações, consulte Como [trabalhar com rotas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway

- Para API obter detalhes, consulte [GetRoutes](#) na Referência de AWS CLI Comandos.

get-stage

O código de exemplo a seguir mostra como usar `get-stage`.

AWS CLI

Para recuperar informações sobre um estágio

O `get-stage` exemplo a seguir exibe informações sobre o `prod` estágio de um API.

```
aws apigatewayv2 get-stage \  
  --api-id a1b2c3d4 \  
  --stage-name prod
```

Saída:

```
{  
  "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": {}  
}
```

Para obter mais informações, consulte Como [trabalhar com estágios HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [GetStage](#) na Referência de AWS CLI Comandos.

get-stages

O código de exemplo a seguir mostra como usar `get-stages`.

AWS CLI

Para recuperar uma lista de estágios

O `get-stages` exemplo a seguir lista todos os estágios API de um.

```
aws apigatewayv2 get-stages \  
  --api-id a1b2c3d4
```

Saída:

```
{  
  "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": {}
}
]
}

```

Para obter mais informações, consulte Como [trabalhar com estágios HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [GetStages](#) na Referência de AWS CLI Comandos.

get-tags

O código de exemplo a seguir mostra como usar get-tags.

AWS CLI

Para recuperar uma lista de tags para um recurso

O get-tags exemplo a seguir lista todas as tags API de um.

```
aws apigatewayv2 get-tags \
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4
```

Saída:

```
{
```

```
"Tags": {
  "owner": "dev-team",
  "environment": "prod"
}
```

Para obter mais informações, consulte Como [marcar seus recursos do API Gateway](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [GetTags](#) na Referência de AWS CLI Comandos.

get-vpc-link

O código de exemplo a seguir mostra como usar `get-vpc-link`.

AWS CLI

Para recuperar informações sobre um link VPC

O `get-vpc-link` exemplo a seguir exibe informações sobre um VPC link.

```
aws apigatewayv2 get-vpc-link \
  --vpc-link-id abcd123
```

Saída:

```
{
  "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"
```

```
}
```

Para obter mais informações, consulte Como [trabalhar com VPC links HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [GetVpcLink](#) na Referência de AWS CLI Comandos.

get-vpc-links

O código de exemplo a seguir mostra como usar `get-vpc-links`.

AWS CLI

Para recuperar uma lista de links VPC

O `get-vpc-links` exemplo a seguir exibe uma lista de todos os VPC links do usuário atual.

```
aws apigatewayv2 get-vpc-links
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte Como [trabalhar com VPC links HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [GetVpcLinks](#) na Referência de AWS CLI Comandos.

import-api

O código de exemplo a seguir mostra como usar `import-api`.

AWS CLI

Para importar um HTTP API

O `import-api` exemplo a seguir cria um HTTP API a partir de um arquivo de definição do Open API 3.0 chamado `api-definition.yaml`.

```

aws apigatewayv2 import-api \
  --body file://api-definition.yaml

```

Conteúdo de `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
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte Como [trabalhar com API definições abertas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [ImportApina](#) Referência de AWS CLI Comandos.

reimport-api

O código de exemplo a seguir mostra como usar `reimport-api`.

AWS CLI

Para reimportar um HTTP API

O `reimport-api` exemplo a seguir atualiza uma existente HTTP API para usar a definição Open API 3.0 especificada em `api-definition.yaml`.

```
aws apigatewayv2 reimport-api \
```

```
--body file://api-definition.yaml \  
--api-id a1b2c3d4
```

Conteúdo de 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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte Como [trabalhar com API definições abertas HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [ReimportApina](#) Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir adiciona uma tag com o nome da chave `Department` e um valor de `Accounting` ao especificadoAPI.

```
aws apigatewayv2 tag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tags Department=Accounting
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos do API Gateway](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Como remover as tags de um recurso

O `untag-resource` exemplo a seguir remove as tags com os nomes de chave `Project` e `Owner` do especificadoAPI.

```
aws apigatewayv2 untag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tag-keys Project Owner
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos do API Gateway](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-api-mapping

O código de exemplo a seguir mostra como usar `update-api-mapping`.

AWS CLI

Para atualizar um API mapeamento

O `update-api-mapping` exemplo a seguir altera um API mapeamento para um nome de domínio personalizado. Como resultado, a base URL usando o nome de domínio personalizado para o estágio especificado API se torna `https://api.example.com/dev`.

```
aws apigatewayv2 update-api-mapping \  
  --api-id a1b2c3d4 \  
  --stage dev \  
  --domain-name api.example.com \  
  --api-mapping-id 0qzs2sy7bh \  
  --api-mapping-key dev
```

Saída:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "0qzs2sy7bh",  
  "ApiMappingKey": "dev"  
  "Stage": "dev"  
}
```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway no Amazon API Gateway Developer Guide](#).

- Para API obter detalhes, consulte [UpdateApiMapping](#) na Referência de AWS CLI Comandos.

update-api

O código de exemplo a seguir mostra como usar `update-api`.

AWS CLI

Para CORS habilitar um HTTP API

O `update-api` exemplo a seguir atualiza a CORS configuração API do especificado para permitir solicitações de `https://www.example.com`.

```
aws apigatewayv2 update-api \  
  --cors-configuration {}
```

```
--api-id a1b2c3d4 \  
--cors-configuration AllowOrigins=https://www.example.com
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Como configurar CORS um HTTP API](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [UpdateApi](#) na Referência de AWS CLI Comandos.

update-authorizer

O código de exemplo a seguir mostra como usar update-authorizer.

AWS CLI

Para atualizar um autorizador

O `update-authorizer` exemplo a seguir altera a fonte de identidade de um JWT autorizador para um cabeçalho chamado `Authorization`.

```
aws apigatewayv2 update-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3 \  
  --identity-source '$request.header.Authorization'
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Controle do acesso HTTP APIs com JWT autorizadores](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [UpdateAuthorizer](#) na Referência de AWS CLI Comandos.

update-deployment

O código de exemplo a seguir mostra como usar `update-deployment`.

AWS CLI

Para alterar a descrição de uma implantação

O `update-deployment` exemplo a seguir atualiza a descrição de uma implantação.

```
aws apigatewayv2 update-deployment \  
  --deployment-id abc123 \  
  --description my deployment
```

```
--api-id a1b2c3d4 \  
--deployment-id abcdef \  
--description 'Manual deployment to fix integration test failures.'
```

Saída:

```
{  
  "AutoDeployed": false,  
  "CreateDate": "2020-02-05T16:21:48+00:00",  
  "DeploymentId": "abcdef",  
  "DeploymentStatus": "DEPLOYED",  
  "Description": "Manual deployment to fix integration test failures."  
}
```

Para obter mais informações, consulte [Desenvolvendo um HTTP API no API Gateway](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [UpdateDeployment](#) na Referência de AWS CLI Comandos.

update-domain-name

O código de exemplo a seguir mostra como usar `update-domain-name`.

AWS CLI

Para atualizar um nome de domínio personalizado

O `update-domain-name` exemplo a seguir especifica um novo ACM certificado para o nome de domínio `api.example.com` personalizado.

```
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
```

Saída:

```
{  
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Configurando um nome de domínio regional personalizado no API Gateway](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [UpdateDomainName](#) na Referência de AWS CLI Comandos.

update-integration

O código de exemplo a seguir mostra como usar update-integration.

AWS CLI

Para atualizar uma integração com o Lambda

O update-integration exemplo a seguir atualiza uma integração existente do AWS Lambda para usar a função Lambda especificada.

```

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

```

Saída:

```

{
  "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
}
```

Para obter mais informações, consulte [Como configurar integrações HTTP APIs e Configurar integrações no Amazon API Gateway WebSocket API Developer Guide](#).

- Para API obter detalhes, consulte [UpdateIntegration](#) na Referência de AWS CLI Comandos.

update-route

O código de exemplo a seguir mostra como usar `update-route`.

AWS CLI

Exemplo 1: Para atualizar a integração de uma rota

O `update-route` exemplo a seguir atualiza a integração de uma rota especificada.

```
aws apigatewayv2 update-route \
  --api-id a1b2c3d4 \
  --route-id a1b2c3 \
  --target integrations/a1b2c6
```

Saída:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
  "RouteId": "a1b2c3",
  "RouteKey": "ANY /pets",
  "Target": "integrations/a1b2c6"
}
```

Exemplo 2: Para adicionar um autorizador a uma rota

O `update-route` exemplo a seguir atualiza a rota especificada para usar um JWT autorizador.

```
aws apigatewayv2 update-route \
```

```
--api-id a1b2c3d4 \  
--route-id a1b2c3 \  
--authorization-type JWT \  
--authorizer-id a1b2c5 \  
--authorization-scopes user.id user.email
```

Saída:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationScopes": [  
    "user.id",  
    "user.email"  
  ],  
  "AuthorizationType": "JWT",  
  "AuthorizerId": "a1b2c5",  
  "OperationName": "GET HTTP",  
  "RequestParameters": {},  
  "RouteId": "a1b2c3",  
  "RouteKey": "GET /pets",  
  "Target": "integrations/a1b2c6"  
}
```

Para obter mais informações, consulte [Controle do acesso HTTP APIs com JWT autorizadores](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [UpdateRoute](#) na Referência de AWS CLI Comandos.

update-stage

O código de exemplo a seguir mostra como usar update-stage.

AWS CLI

Para configurar a limitação personalizada

O update-stage exemplo a seguir configura a limitação personalizada para o estágio e a rota especificados de um API

```
aws apigatewayv2 update-stage \  
--api-id a1b2c3d4 \  
--stage-name prod \  
--route-id a1b2c3 \  
--limit 100 \  
--limit-type PERCENTAGE
```

```
--stage-name dev \  
--route-settings '{"GET /pets":  
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

Saída:

```
{  
  "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": {}  
}
```

Para obter mais informações, consulte [Protegendo seu HTTP API](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [UpdateStage](#) na Referência de AWS CLI Comandos.

update-vpc-link

O código de exemplo a seguir mostra como usar `update-vpc-link`.

AWS CLI

Para atualizar um VPC link

O `update-vpc-link` exemplo a seguir atualiza o nome de um VPC link. Depois de criar um VPC link, você não poderá alterar seus grupos de segurança ou sub-redes.

```
aws apigatewayv2 update-vpc-link \  
--vpc-link-id abcd123 \  

```

```
--name MyUpdatedVpcLink
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte Como [trabalhar com VPC links HTTP APIs](#) no Guia do desenvolvedor do Amazon API Gateway.

- Para API obter detalhes, consulte [UpdateVpcLink](#) na Referência de AWS CLI Comandos.

APIAPIExemplos de gerenciamento de gateway usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o API Gateway ManagementAPI.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-connection

O código de exemplo a seguir mostra como usar `delete-connection`.

AWS CLI

Para excluir uma WebSocket conexão

O `delete-connection` exemplo a seguir desconecta um cliente do especificado WebSocket API.

```
aws apigatewaymanagementapi delete-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Este comando não produz saída.

Para obter mais informações, consulte [Use os comandos @connections em seu serviço de back-end](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [DeleteConnection](#) na Referência de AWS CLI Comandos.

get-connection

O código de exemplo a seguir mostra como usar `get-connection`.

AWS CLI

Para obter informações sobre uma WebSocket conexão

O `get-connection` exemplo a seguir descreve uma conexão com o especificado WebSocket API.

```
aws apigatewaymanagementapi get-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Saída:

```
{  
  "ConnectedAt": "2020-04-30T20:10:33.236Z",
```

```
"Identity": {
  "SourceIp": "192.0.2.1"
},
"LastActiveAt": "2020-04-30T20:10:42.997Z"
}
```

Para obter mais informações, consulte [Use os comandos @connections em seu serviço de back-end](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [GetConnection](#) na Referência de AWS CLI Comandos.

post-to-connection

O código de exemplo a seguir mostra como usar post-to-connection.

AWS CLI

Para enviar dados para uma WebSocket conexão

O post-to-connection exemplo a seguir envia uma mensagem para um cliente conectado ao especificado WebSocket API.

```
aws apigatewaymanagementapi post-to-connection \
  --connection-id L0SM9c0FvHcCIhw= \
  --data "Hello from API Gateway!" \
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Este comando não produz saída.

Para obter mais informações, consulte [Use os comandos @connections em seu serviço de back-end](#) no Amazon API Gateway Developer Guide.

- Para API obter detalhes, consulte [PostToConnection](#) na Referência de AWS CLI Comandos.

Exemplos de App Mesh usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o App Mesh.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-mesh

O código de exemplo a seguir mostra como usar `create-mesh`.

AWS CLI

Exemplo 1: Para criar uma nova malha de serviços

O `create-mesh` exemplo a seguir cria uma malha de serviços.

```
aws appmesh create-mesh \  
  --mesh-name app1
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Exemplo 2: Para criar uma nova malha de serviços com várias tags

O `create-mesh` exemplo a seguir cria uma malha de serviços com várias tags.

```
aws appmesh create-mesh \  
  --mesh-name app2 \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Service Meshes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [CreateMesh](#) na Referência de AWS CLI Comandos.

create-route

O código de exemplo a seguir mostra como usar `create-route`.

AWS CLI

Para criar uma nova RPC rota g

O `create-route` exemplo a seguir usa um arquivo JSON de entrada para criar uma RPC rota g. GRPCo tráfego que tem metadados que começam com 123 é roteado para um nó virtual chamado. `serviceBgrpc` Se houver TCP falhas específicas de g `RPCHTTP`,, ou ao tentar se comunicar com o destino da rota, a rota será repetida três vezes. Há um atraso de 15 segundos entre cada nova tentativa.


```
aws appmesh create-route \  
  --cli-input-json file://create-route-grpc.json
```

Conteúdo de 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"  
}
```

Saída:

```
{  
  "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"
}
}

```

Para criar uma rota nova HTTP ou HTTP /2

O `create-route` exemplo a seguir usa um arquivo JSON de entrada para criar uma rota HTTP /2. Para criar uma HTTP rota, substitua `http2Route` por `httpRoute` under `spec`. Todo o tráfego HTTP /2 endereçado a qualquer URL prefixo que tenha um valor de cabeçalho que comece com 123 é roteado para um nó virtual chamado 2. `serviceBhttp` Se houver falhas específicas HTTP ou TCP falhas ao tentar se comunicar com o destino da rota, a rota será repetida três vezes. Há um atraso de 15 segundos entre cada nova tentativa.

```

aws appmesh create-route \
  --cli-input-json file://create-route-http2.json

```

Conteúdo de `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"
}
```

Saída:

```
{
  "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"
}
}

```

Para criar uma nova TCP rota

O `create-route` exemplo a seguir usa um arquivo JSON de entrada para criar uma TCP rota. 75 por cento do tráfego é roteado para um nó virtual chamado `serviceBtcp` e 25 por cento do tráfego é roteado para um nó virtual chamado `serviceBv2tcp`. Especificar pesos diferentes para alvos diferentes é uma forma eficaz de implantar uma nova versão de um aplicativo. Você pode ajustar os pesos para que, eventualmente, 100% de todo o tráfego seja roteado para um destino que tenha a nova versão de um aplicativo.

```

aws appmesh create-route \
  --cli-input-json file://create-route-tcp.json

```

Conteúdo `create-route-tcp` do `do.json`:

```

{
  "meshName": "apps",
  "routeName": "tcpRoute",
  "spec": {
    "priority": 300,
    "tcpRoute": {
      "action": {
        "weightedTargets": [
          {
            "virtualNode": "serviceBtcp",

```

```

        "weight": 75
      },
      {
        "virtualNode": "serviceBv2tcp",
        "weight": 25
      }
    ]
  }
},
"virtualRouterName": "serviceBtcp"
}

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Routes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [CreateRoute](#) na Referência de AWS CLI Comandos.

create-virtual-gateway

O código de exemplo a seguir mostra como usar `create-virtual-gateway`.

AWS CLI

Para criar um novo gateway virtual

O `create-virtual-gateway` exemplo a seguir usa um arquivo JSON de entrada para criar um gateway virtual com um ouvinte para HTTP usar a porta 9080.

```

aws appmesh create-virtual-gateway \
  --mesh-name meshName \
  --virtual-gateway-name virtualGatewayName \
  --cli-input-json file://create-virtual-gateway.json

```

Conteúdo de `create-virtual-gateway.json`:

```

{
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 9080,
          "protocol": "http"
        }
      }
    ]
  }
}

```


Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Virtual Gateways](#) no Guia do Usuário do AWS App Mesh.

- Para API obter detalhes, consulte [CreateVirtualGateway](#) na Referência de AWS CLI Comandos.

create-virtual-node

O código de exemplo a seguir mostra como usar `create-virtual-node`.

AWS CLI

Exemplo 1: Para criar um novo nó virtual usado DNS para descoberta

O `create-virtual-node` exemplo a seguir usa um arquivo JSON de entrada para criar um nó virtual usado DNS para descoberta de serviços.

```
aws appmesh create-virtual-node \  
  --cli-input-json file://create-virtual-node-dns.json
```

Conteúdo de `create-virtual-node-dns.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ],  
    "serviceDiscovery": {  
      "dns": {  
        "hostname": "serviceBv1.svc.cluster.local"  
      }  
    }  
  },  
  "virtualNodeName": "vnServiceBv1"  
}
```

Saída:

```
{  
  "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"
}

```

Exemplo 2: Para criar um novo nó virtual que usa o AWS Cloud Map para descoberta

O `create-virtual-node` exemplo a seguir usa um arquivo JSON de entrada para criar um nó virtual que usa o AWS Cloud Map para descoberta de serviços.

```

aws appmesh create-virtual-node \
  --cli-input-json file://create-virtual-node-cloud-map.json

```

Conteúdo de `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"
}

```

Saída:

```

{
  "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"
  }
}
```

Para obter mais informações, consulte [Virtual Nodes](#) no Guia do Usuário do AWS App Mesh.

- Para API obter detalhes, consulte [CreateVirtualNode](#) na Referência de AWS CLI Comandos.

create-virtual-router

O código de exemplo a seguir mostra como usar `create-virtual-router`.

AWS CLI

Para criar um novo roteador virtual

O `create-virtual-router` exemplo a seguir usa um arquivo JSON de entrada para criar um roteador virtual com um ouvinte para HTTP usar a porta 80.

```
aws appmesh create-virtual-router \  
  --cli-input-json file://create-virtual-router.json
```

Conteúdo de create-virtual-router.json:

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ]  
  },  
  "virtualRouterName": "vrServiceB"  
}
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Roteadores virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [CreateVirtualRouter](#) na Referência de AWS CLI Comandos.

create-virtual-service

O código de exemplo a seguir mostra como usar `create-virtual-service`.

AWS CLI

Exemplo 1: Para criar um novo serviço virtual com um provedor de nós virtuais

O `create-virtual-service` exemplo a seguir usa um arquivo JSON de entrada para criar um serviço virtual com um provedor de nó virtual.

```

aws appmesh create-virtual-service \
  --cli-input-json file://create-virtual-service-virtual-node.json

```

Conteúdo de `create-virtual-service-virtual-node.json`:

```

{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualNode": {
        "virtualNodeName": "vnServiceA"
      }
    }
  },
  "virtualServiceName": "serviceA.svc.cluster.local"
}

```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Virtual Node](#) no Guia do Usuário do AWS App Mesh.

Exemplo 2: Para criar um novo serviço virtual com um provedor de roteador virtual

O `create-virtual-service` exemplo a seguir usa um arquivo JSON de entrada para criar um serviço virtual com um provedor de roteador virtual.

```
aws appmesh create-virtual-service \
  --cli-input-json file://create-virtual-service-virtual-router.json
```

Conteúdo de `create-virtual-service-virtual-router.json`:

```
{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualRouter": {
```



```

        "virtualRouterName": "vrServiceB"
    }
}
},
"virtualServiceName": "serviceB.svc.cluster.local"
}

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte Virtual Services < https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html > no Guia do usuário do App Mesh AWS

- Para API obter detalhes, consulte [CreateVirtualService](#) na Referência de AWS CLI Comandos.

delete-mesh

O código de exemplo a seguir mostra como usar delete-mesh.

AWS CLI

Para excluir uma malha de serviços

O `delete-mesh` exemplo a seguir exclui a malha de serviços especificada.

```
aws appmesh delete-mesh \  
  --mesh-name app1
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Service Meshes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DeleteMesh](#) na Referência de AWS CLI Comandos.

delete-route

O código de exemplo a seguir mostra como usar `delete-route`.

AWS CLI

Para excluir uma rota

O `delete-route` exemplo a seguir exclui a rota especificada.

```
aws appmesh delete-route \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB \  
  --route-name toVnServiceB-weighted
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Routes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DeleteRoute](#) na Referência de AWS CLI Comandos.

delete-virtual-node

O código de exemplo a seguir mostra como usar delete-virtual-node.

AWS CLI

Para excluir um nó virtual

O delete-virtual-node exemplo a seguir exclui o nó virtual especificado.

```

aws appmesh delete-virtual-node \
  --mesh-name app1 \
  --virtual-node-name vnServiceBv2

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Virtual Nodes](#) no Guia do Usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DeleteVirtualNode](#) na Referência de AWS CLI Comandos.

delete-virtual-router

O código de exemplo a seguir mostra como usar `delete-virtual-router`.

AWS CLI

Para excluir um roteador virtual

O `delete-virtual-router` exemplo a seguir exclui o roteador virtual especificado.

```

aws appmesh delete-virtual-router \
  --mesh-name app1 \
  --virtual-router-name vrServiceB

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Roteadores virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DeleteVirtualRouter](#) na Referência de AWS CLI Comandos.

delete-virtual-service

O código de exemplo a seguir mostra como usar delete-virtual-service.

AWS CLI

Para excluir um serviço virtual

O delete-virtual-service exemplo a seguir exclui o serviço virtual especificado.

```

aws appmesh delete-virtual-service \
  --mesh-name app1 \
  --virtual-service-name serviceB.svc.cluster.local

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Serviço virtual](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DeleteVirtualService](#) na Referência de AWS CLI Comandos.

describe-mesh

O código de exemplo a seguir mostra como usar `describe-mesh`.

AWS CLI

Para descrever uma malha de serviços

O `describe-mesh` exemplo a seguir retorna detalhes sobre a malha de serviços especificada.

```

aws appmesh describe-mesh \
  --mesh-name app1

```

Saída:

```

{
  "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"
    }
}
}

```

Para obter mais informações, consulte [Service Meshes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DescribeMesh](#) na Referência de AWS CLI Comandos.

describe-route

O código de exemplo a seguir mostra como usar `describe-route`.

AWS CLI

Para descrever uma rota

O `describe-route` exemplo a seguir retorna detalhes sobre a rota especificada.

```

aws appmesh describe-route \
  --mesh-name app1 \
  --virtual-router-name vrServiceB \
  --route-name toVnServiceB-weighted

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Routes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DescribeRoute](#) na Referência de AWS CLI Comandos.

describe-virtual-node

O código de exemplo a seguir mostra como usar `describe-virtual-node`.

AWS CLI

Para descrever um nó virtual

O `describe-virtual-node` exemplo a seguir retorna detalhes sobre o nó virtual especificado.

```

aws appmesh describe-virtual-node \
  --mesh-name app1 \
  --virtual-node-name vnServiceBv1

```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Virtual Nodes](#) no Guia do Usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DescribeVirtualNode](#) na Referência de AWS CLI Comandos.

describe-virtual-router

O código de exemplo a seguir mostra como usar `describe-virtual-router`.

AWS CLI

Para descrever um roteador virtual

O `describe-virtual-router` exemplo a seguir retorna detalhes sobre o roteador virtual especificado.

```
aws appmesh describe-virtual-router \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Roteadores virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DescribeVirtualRouter](#) na Referência de AWS CLI Comandos.

describe-virtual-service

O código de exemplo a seguir mostra como usar `describe-virtual-service`.

AWS CLI

Para descrever um serviço virtual

O `describe-virtual-service` exemplo a seguir retorna detalhes sobre o serviço virtual especificado.

```
aws appmesh describe-virtual-service \  
  --mesh-name app1 \  
  --virtual-service-name serviceB.svc.cluster.local
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Serviços virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [DescribeVirtualService](#) na Referência de AWS CLI Comandos.

list-meshes

O código de exemplo a seguir mostra como usar `list-meshes`.

AWS CLI

Para listar malhas de serviço

O `list-meshes` exemplo a seguir lista todas as malhas de serviço na AWS região atual.

```
aws appmesh list-meshes
```

Saída:

```
{
  "meshes": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
      "meshName": "app1"
    }
  ]
}
```

Para obter mais informações, consulte [Service Meshes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [ListMeshes](#) na Referência de AWS CLI Comandos.

list-routes

O código de exemplo a seguir mostra como usar `list-routes`.

AWS CLI

Para listar rotas

O `list-routes` exemplo a seguir lista todas as rotas para o roteador virtual especificado.

```
aws appmesh list-routes \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB
```

Saída:

```
{  
  "routes": [  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB/route/toVnServiceB",  
      "meshName": "app1",  
      "routeName": "toVnServiceB-weighted",  
      "virtualRouterName": "vrServiceB"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Routes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [ListRoutes](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista todas as tags atribuídas ao recurso especificado.

```
aws appmesh list-tags-for-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

Saída:

```
{
```

```
"tags": [
  {
    "key": "key1",
    "value": "value1"
  },
  {
    "key": "key2",
    "value": "value2"
  },
  {
    "key": "key3",
    "value": "value3"
  }
]
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-virtual-nodes

O código de exemplo a seguir mostra como usar `list-virtual-nodes`.

AWS CLI

Para listar nós virtuais

O `list-virtual-nodes` exemplo a seguir lista todos os nós virtuais na malha de serviços especificada.

```
aws appmesh list-virtual-nodes \
  --mesh-name app1
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações, consulte [Virtual Nodes](#) no Guia do Usuário do AWS App Mesh.

- Para API obter detalhes, consulte [ListVirtualNodes](#) na Referência de AWS CLI Comandos.

list-virtual-routers

O código de exemplo a seguir mostra como usar `list-virtual-routers`.

AWS CLI

Para listar roteadores virtuais

O `list-virtual-routers` exemplo a seguir lista todos os roteadores virtuais na malha de serviço especificada.

```
aws appmesh list-virtual-routers \
  --mesh-name app1
```

Saída:

```
{
  "virtualRouters": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
      "meshName": "app1",
      "virtualRouterName": "vrServiceB"
    }
  ]
}
```

Para obter mais informações, consulte [Roteadores virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [ListVirtualRouters](#) na Referência de AWS CLI Comandos.

list-virtual-services

O código de exemplo a seguir mostra como usar `list-virtual-services`.

AWS CLI

Para listar serviços virtuais

O `list-virtual-services` exemplo a seguir lista todos os serviços virtuais na malha de serviços especificada.

```
aws appmesh list-virtual-services \
  --mesh-name app1
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Serviços virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [ListVirtualServices](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir adiciona a tag `key1` com o valor `value1` ao recurso especificado.

```
aws appmesh tag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tags key=key1,value=value1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para desmarcar um recurso

O `untag-resource` exemplo a seguir remove uma tag com a chave `key1` do recurso especificado.

```
aws appmesh untag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tag-keys key1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-mesh

O código de exemplo a seguir mostra como usar `update-mesh`.

AWS CLI

Para atualizar uma malha de serviços

O `update-mesh` exemplo a seguir usa um arquivo JSON de entrada para atualizar uma malha de serviços para permitir que todo o tráfego de saída externo seja encaminhado pelo proxy Envoy intocado.

```
aws appmesh update-mesh \  
  --cli-input-json file://update-mesh.json
```

Conteúdo de `update-mesh.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "egressFilter": {  
      "type": "ALLOW_ALL"  
    }  
  }  
}
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Service Meshes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [UpdateMesh](#) na Referência de AWS CLI Comandos.

update-route

O código de exemplo a seguir mostra como usar update-route.

AWS CLI

Para atualizar uma rota

O update-route exemplo a seguir usa um arquivo JSON de entrada para atualizar os pesos de uma rota.

```
aws appmesh update-route \  
  --cli-input-json file://update-route-weighted.json
```

Conteúdo de 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"  
}
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Routes](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [UpdateRoute](#) na Referência de AWS CLI Comandos.

update-virtual-node

O código de exemplo a seguir mostra como usar `update-virtual-node`.

AWS CLI

Para atualizar um nó virtual

O `update-virtual-node` exemplo a seguir usa um arquivo JSON de entrada para adicionar uma verificação de saúde a um nó virtual.

```
aws appmesh update-virtual-node \  
  --cli-input-json file://update-virtual-node.json
```

Conteúdo de `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"
}
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Virtual Nodes](#) no Guia do Usuário do AWS App Mesh.

- Para API obter detalhes, consulte [UpdateVirtualNode](#) na Referência de AWS CLI Comandos.

update-virtual-router

O código de exemplo a seguir mostra como usar `update-virtual-router`.

AWS CLI

Para atualizar um roteador virtual

O `update-virtual-router` exemplo a seguir usa um arquivo JSON de entrada para atualizar uma porta de ouvinte de roteador virtual.

```
aws appmesh update-virtual-router \
  --cli-input-json file://update-virtual-router.json
```

Conteúdo de `update-virtual-router.json`:

```
{
  "meshName": "app1",
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 8080,
          "protocol": "http"
        }
      }
    ]
  },
  "virtualRouterName": "vrServiceB"
}
```

Saída:

```
{
```



```
"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"
}
}
```

Para obter mais informações, consulte [Roteadores virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [UpdateVirtualRouter](#) na Referência de AWS CLI Comandos.

update-virtual-service

O código de exemplo a seguir mostra como usar `update-virtual-service`.

AWS CLI

Para atualizar um serviço virtual

O `update-virtual-service` exemplo a seguir usa um arquivo JSON de entrada para atualizar um serviço virtual para usar um provedor de roteador virtual.

```
aws appmesh update-virtual-service \  
--cli-input-json file://update-virtual-service.json
```

Conteúdo de `update-virtual-service.json`:

```
{  
  "meshName": "app1",  
  "spec": {  
    "provider": {  
      "virtualRouter": {  
        "virtualRouterName": "vrServiceA"  
      }  
    }  
  },  
  "virtualServiceName": "serviceA.svc.cluster.local"  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Serviços virtuais](#) no Guia do usuário do AWS App Mesh.

- Para API obter detalhes, consulte [UpdateVirtualService](#) na Referência de AWS CLI Comandos.

Exemplos de App Runner usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o App Runner.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-custom-domain

O código de exemplo a seguir mostra como usar `associate-custom-domain`.

AWS CLI

Para associar um nome de domínio e o subdomínio `www` a um serviço

O `associate-custom-domain` exemplo a seguir associa um nome de domínio personalizado que você controla a um serviço App Runner. O nome do domínio é o domínio raiz `example.com`, incluindo o subdomínio de casos especiais. `www.example.com`

```
aws apprunner associate-custom-domain \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "DomainName": "example.com",
  "EnableWWWSubdomain": true
}
```

Saída:

```
{
  "CustomDomain": {
    "CertificateValidationRecords": [
      {
        "Name": "_70d3f50a94f7c72dc28784cf55db2f6b.example.com",
        "Status": "PENDING_VALIDATION",
        "Type": "CNAME",
        "Value": "_1270c137383c6307b6832db02504c4b0.bsgbmzkfwj.acm-
validations.aws."
      },
      {
        "Name": "_287870d3f50a94f7c72dc4cf55db2f6b.www.example.com",
        "Status": "PENDING_VALIDATION",
        "Type": "CNAME",
        "Value": "_832db01270c137383c6307b62504c4b0.mzkbsgbfwj.acm-
validations.aws."
      }
    ],
    "DomainName": "example.com",
    "EnableWWWSubdomain": true,
    "Status": "CREATING"
  },
  "DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

- Para API obter detalhes, consulte [AssociateCustomDomain](#) na Referência de AWS CLI Comandos.

create-auto-scaling-configuration

O código de exemplo a seguir mostra como usar create-auto-scaling-configuration.

AWS CLI

Para criar uma configuração de escalonamento automático de alta disponibilidade

O `create-auto-scaling-configuration` exemplo a seguir cria uma configuração de escalonamento automático otimizada para alta disponibilidade, definindo como `MinSize` 5. Com essa configuração, o App Runner tenta distribuir suas instâncias de serviço pelo maior número possível de zonas de disponibilidade, até cinco, dependendo da AWS região.

A chamada retorna um `AutoScalingConfiguration` objeto com as outras configurações definidas como padrão. No exemplo, essa é a primeira chamada para criar uma configuração chamada `high-availability`. A revisão está definida como 1 e é a revisão mais recente.

```
aws apprunner create-auto-scaling-configuration \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "AutoScalingConfigurationName": "high-availability",  
  "MinSize": 5  
}
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [CreateAutoScalingConfiguration](#) na Referência de AWS CLI Comandos.

create-connection

O código de exemplo a seguir mostra como usar `create-connection`.

AWS CLI

Para criar uma GitHub conexão

O `create-connection` exemplo a seguir cria uma conexão com um repositório GitHub de código privado. O status da conexão após uma chamada bem-sucedida é `PENDING_HANDSHAKE`. Isso ocorre porque um handshake de autenticação com o provedor ainda não aconteceu. Conclua o handshake usando o console do App Runner.

```
aws apprunner create-connection \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "ConnectionName": "my-github-connection",  
  "ProviderType": "GITHUB"  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Gerenciando conexões do App Runner](#) no Guia do desenvolvedor do AWS App Runner.

- Para API obter detalhes, consulte [CreateConnection](#) na Referência de AWS CLI Comandos.

create-service

O código de exemplo a seguir mostra como usar `create-service`.

AWS CLI

Exemplo 1: Para criar um serviço de repositório de código-fonte

O `create-service` exemplo a seguir cria um serviço App Runner com base em um repositório de código-fonte do Python.

```
aws apprunner create-service \  
  --cli-input-json file://input.json
```

Conteúdo de `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"
}
}

```

Saída:

```

{
  "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"
}
}
}

```

Exemplo 2: Para criar um serviço de repositório de código-fonte

O `create-service` exemplo a seguir cria um serviço App Runner com base em um repositório de código-fonte do Python.

```

aws apprunner create-service \
  --cli-input-json file://input.json

```

Conteúdo de `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"
    }
}

```

Saída:

```

{
  "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": {

```



```

    "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"
  }
}

```

Saída:

```

{
  "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"
    }
}

```

- Para API obter detalhes, consulte [CreateService](#) na Referência de AWS CLI Comandos.

delete-auto-scaling-configuration

O código de exemplo a seguir mostra como usar delete-auto-scaling-configuration.

AWS CLI

Exemplo 1: Para excluir a revisão ativa mais recente de uma configuração de auto scaling

O delete-auto-scaling-configuration exemplo a seguir exclui a revisão ativa mais recente de uma configuração de escalonamento automático do App Runner. Para excluir a última revisão ativa, especifique um Amazon Resource Name (ARN) que termine com o nome da configuração, sem o componente de revisão.

No exemplo, existem duas revisões antes dessa ação. Portanto, a revisão 2 (a mais recente) é excluída. No entanto, agora aparece "Latest": false, porque, depois de ser excluída, não é mais a última revisão ativa.

```
aws apprunner delete-auto-scaling-configuration \
  --cli-input-json file://input.json
```

Conteúdo de input.json:

```
{
```

```
"AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

Saída:

```
{
  "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
  }
}
```

Exemplo 2: Para excluir uma revisão específica de uma configuração de auto scaling

O `delete-auto-scaling-configuration` exemplo a seguir exclui uma revisão específica de uma configuração de escalonamento automático do App Runner. Para excluir uma revisão específica, especifique uma ARN que inclua o número da revisão.

No exemplo, existem várias revisões antes dessa ação. A ação exclui a revisão1.

```
aws apprunner delete-auto-scaling-configuration \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [DeleteAutoScalingConfiguration](#) na Referência de AWS CLI Comandos.

delete-connection

O código de exemplo a seguir mostra como usar delete-connection.

AWS CLI

Para excluir uma conexão

O delete-connection exemplo a seguir exclui uma conexão do App Runner. O status da conexão após uma chamada bem-sucedida é DELETED. Isso ocorre porque a conexão não está mais disponível.

```
aws apprunner delete-connection \
  --cli-input-json file://input.json
```

Conteúdo de input.json:

```
{
  "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-
connection"
}
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [DeleteConnection](#) na Referência de AWS CLI Comandos.

delete-service

O código de exemplo a seguir mostra como usar delete-service.

AWS CLI

Como excluir um serviço

O delete-service exemplo a seguir exclui um serviço App Runner.

```
aws apprunner delete-service \
  --cli-input-json file://input.json
```

Conteúdo de input.json:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Saída:

```
{
  "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"
    }
  }
}

```

- Para API obter detalhes, consulte [DeleteService](#) na Referência de AWS CLI Comandos.

describe-auto-scaling-configuration

O código de exemplo a seguir mostra como usar `describe-auto-scaling-configuration`.

AWS CLI

Exemplo 1: Para descrever a última revisão ativa de uma configuração de auto scaling

O `describe-auto-scaling-configuration` exemplo a seguir apresenta uma descrição da revisão ativa mais recente de uma configuração de escalonamento automático do App Runner. Para descrever a última revisão ativa, especifique uma ARN que termine com o nome da configuração, sem o componente de revisão.

No exemplo, existem duas revisões. Portanto, a revisão 2 (a mais recente) é descrita. O objeto resultante é exibido `"Latest": true`.

```
aws apprunner describe-auto-scaling-configuration \  
--cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability"  
}
```

Saída:

```
{  
  "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  
  }  
}
```

```
}
```

Exemplo 2: Para descrever uma revisão específica de uma configuração de auto scaling

O `describe-auto-scaling-configuration` exemplo a seguir mostra uma descrição de uma revisão específica de uma configuração de escalonamento automático do App Runner. Para descrever uma revisão específica, especifique uma ARN que inclua o número da revisão.

No exemplo, existem várias revisões e a revisão 1 é consultada. O objeto resultante é exibido `"Latest": false`.

```
aws apprunner describe-auto-scaling-configuration \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability/1"  
}
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [DescribeAutoScalingConfiguration](#) na Referência de AWS CLI Comandos.

describe-custom-domains

O código de exemplo a seguir mostra como usar `describe-custom-domains`.

AWS CLI

Para obter descrições de nomes de domínio personalizados associados a um serviço

O `describe-custom-domains` exemplo a seguir mostra as descrições e o status dos nomes de domínio personalizados associados a um serviço App Runner.

```
aws apprunner describe-custom-domains \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "DomainName": "example.com",  
  "EnableWWWSubdomain": true  
}
```

Saída:

```
{  
  "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"
}

```

- Para API obter detalhes, consulte [DescribeCustomDomains](#) na Referência de AWS CLI Comandos.

describe-service

O código de exemplo a seguir mostra como usar `describe-service`.

AWS CLI

Para descrever um serviço

O `describe-service` exemplo a seguir obtém uma descrição de um serviço do App Runner.

```
aws apprunner describe-service \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Saída:

```
{
  "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"
  }
}
}
}

```

- Para API obter detalhes, consulte [DescribeService](#) na Referência de AWS CLI Comandos.

disassociate-custom-domain

O código de exemplo a seguir mostra como usar `disassociate-custom-domain`.

AWS CLI

Para desassociar um nome de domínio de um serviço

O `disassociate-custom-domain` exemplo a seguir desassocia o domínio `example.com` de um serviço App Runner. A chamada também desassocia o subdomínio `www.example.com` associado ao domínio raiz.

```
aws apprunner disassociate-custom-domain \
  --cli-input-json file:///input.json
```

Conteúdo de `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "DomainName": "example.com"
}
```

Saída:

```
{
  "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"
}

```

- Para API obter detalhes, consulte [DisassociateCustomDomain](#) na Referência de AWS CLI Comandos.

list-auto-scaling-configurations

O código de exemplo a seguir mostra como usar `list-auto-scaling-configurations`.

AWS CLI

Para obter uma lista paginada das configurações de escalonamento automático do App Runner

O `list-auto-scaling-configurations` exemplo a seguir lista todas as configurações de escalonamento automático do App Runner em sua conta. AWS Até cinco configurações de auto scaling estão listadas em cada resposta. `AutoScalingConfigurationName` e `LatestOnly` não são especificados. Seus padrões fazem com que a revisão mais recente de todas as configurações ativas seja listada.

Neste exemplo, a resposta inclui dois resultados e não há outros, então não `NextToken` é retornado.


```
aws apprunner list-auto-scaling-configurations \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "MaxResults": 5
}
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListAutoScalingConfigurations](#) na Referência de AWS CLI Comandos.

list-connections

O código de exemplo a seguir mostra como usar `list-connections`.

AWS CLI

Exemplo 1: Para listar todas as conexões

O `list-connections` exemplo a seguir lista todas as conexões do App Runner na AWS conta.

```
aws apprunner list-connections
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: Para listar uma conexão pelo nome

O `list-connections` exemplo a seguir lista uma conexão pelo nome.

```
aws apprunner list-connections \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ConnectionName": "my-github-org-connection"
}
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListConnections](#) na Referência de AWS CLI Comandos.

list-operations

O código de exemplo a seguir mostra como usar `list-operations`.

AWS CLI

Para listar as operações que ocorreram em um serviço

O `list-operations` exemplo a seguir lista todas as operações que ocorreram em um serviço App Runner até o momento. Neste exemplo, o serviço é novo e somente uma única operação do tipo `CREATE_SERVICE` ocorreu.

```
aws apprunner list-operations \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Saída:

```
{
```

```

    "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
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ListOperations](#) na Referência de AWS CLI Comandos.

list-services

O código de exemplo a seguir mostra como usar `list-services`.

AWS CLI

Para obter uma lista paginada dos serviços do App Runner

O `list-services` exemplo a seguir lista todos os serviços do App Runner na AWS conta. Até dois serviços são listados em cada resposta. Este exemplo mostra a primeira solicitação. A resposta inclui dois resultados e um token que pode ser usado na próxima solicitação. Quando uma resposta subsequente não inclui um token, todos os serviços são listados.

```

aws apprunner list-services \
  --cli-input-json file://input.json

```

Conteúdo de `input.json`:

```

{
  "MaxResults": 2
}

```

Saída:

```

{

```

```

    "NextToken":
    "eyJJdDhXN0b21lckFjY291bnRjZCI6IjI3MDIwNTQwMjg0NSIsI1NlcnZpY2VTdGF0dXNDb2RlIjojoiUFJpVklTSU90SU
    "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"
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ListServices](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags associadas a um serviço App Runner

O `list-tags-for-resource` exemplo a seguir lista todas as tags associadas a um serviço App Runner.

```
aws apprunner list-tags-for-resource \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Saída:

```
{
  "Tags": [
    {
      "Key": "Department",
      "Value": "Retail"
    },
    {
      "Key": "CustomerId",
      "Value": "56439872357912"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

pause-service

O código de exemplo a seguir mostra como usar `pause-service`.

AWS CLI

Para pausar um serviço

O `pause-service` exemplo a seguir pausa um serviço App Runner.

```
aws apprunner pause-service \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

Saída:

```
{
  "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"
      }
    }
  }
}

```

- Para API obter detalhes, consulte [PauseService](#) na Referência de AWS CLI Comandos.

resume-service

O código de exemplo a seguir mostra como usar `resume-service`.

AWS CLI

Para retomar um serviço

O `resume-service` exemplo a seguir retoma um serviço App Runner.

```

aws apprunner resume-service \
  --cli-input-json file://input.json

```

Conteúdo de `input.json`:

```

{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

Saída:

```

{
  "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"
}
}
}

```

- Para API obter detalhes, consulte [ResumeService](#) na Referência de AWS CLI Comandos.

start-deployment

O código de exemplo a seguir mostra como usar `start-deployment`.

AWS CLI

Para iniciar uma implantação manual

O `start-deployment` exemplo a seguir executa uma implantação manual em um serviço App Runner.

```
aws apprunner start-deployment \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

Saída:

```
{  
  "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"  
}
```

- Para API obter detalhes, consulte [StartDeployment](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um serviço App Runner

O `tag-resource` exemplo a seguir adiciona duas tags a um serviço App Runner.

```
aws apprunner tag-resource \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "Tags": [
    {
      "Key": "Department",
      "Value": "Retail"
    },
    {
      "Key": "CustomerId",
      "Value": "56439872357912"
    }
  ]
}
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um serviço App Runner

O `untag-resource` exemplo a seguir remove duas tags de um serviço App Runner.

```
aws apprunner untag-resource \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "TagKeys": [
    "Department",
    "CustomerId"
  ]
}
```

```
}
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-service

O código de exemplo a seguir mostra como usar `update-service`.

AWS CLI

Para atualizar o tamanho da memória

O `update-service` exemplo a seguir atualiza o tamanho da memória das instâncias (unidades de escalabilidade) de um serviço App Runner para 2048 MiB.

Quando a chamada é bem-sucedida, o App Runner inicia um processo de atualização assíncrona. A Service estrutura retornada pela chamada reflete o novo valor de memória que está sendo aplicado por essa chamada.

```
aws apprunner update-service \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "InstanceConfiguration": {  
    "Memory": "4 GB"  
  }  
}
```

Saída:

```
{  
  "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"
}
}
```

- Para API obter detalhes, consulte [UpdateService](#) na Referência de AWS CLI Comandos.

AWS AppConfig exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS AppConfig.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-application

O código de exemplo a seguir mostra como usar create-application.

AWS CLI

Para criar um aplicativo

O create-application exemplo a seguir cria um aplicativo em AWS AppConfig.

```
aws appconfig create-application \  
  --name "example-application" \  
  --description "An application used for creating an example."
```

Saída:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [CreateApplication](#) na Referência de AWS CLI Comandos.

create-configuration-profile

O código de exemplo a seguir mostra como usar create-configuration-profile.

AWS CLI

Para criar um perfil de configuração

O create-configuration-profile exemplo a seguir cria um perfil de configuração usando uma configuração armazenada no Parameter Store, um recurso do 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"
```

Saída:

```
{  
  "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  
}
```

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [CreateConfigurationProfile](#) na Referência de AWS CLI Comandos.

create-environment

O código de exemplo a seguir mostra como usar create-environment.

AWS CLI

Para criar um ambiente

O `create-environment` exemplo a seguir cria um AWS AppConfig ambiente chamado `Example-Environment` usando o aplicativo que você criou usando `create-application`.

```
aws appconfig create-environment \  
  --application-id "339ohji" \  
  --name "Example-Environment"
```

Saída:

```
{  
  "ApplicationId": "339ohji",  
  "Description": null,  
  "Id": "54j1r29",  
  "Monitors": null,  
  "Name": "Example-Environment",  
  "State": "ReadyForDeployment"  
}
```

Para obter mais informações, consulte [Etapa 2: Criação de um ambiente](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [CreateEnvironment](#) na Referência de AWS CLI Comandos.

`create-extension-association`

O código de exemplo a seguir mostra como usar `create-extension-association`.

AWS CLI

Para criar uma associação de extensão

O `create-extension-association` exemplo a seguir cria uma nova associação de extensão em 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
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [CreateExtensionAssociation](#) na Referência de AWS CLI Comandos.

create-extension

O código de exemplo a seguir mostra como usar create-extension.

AWS CLI

Para criar uma extensão

O create-extension exemplo a seguir cria uma nova extensão em 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}
```

Saída:

```
{
  "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
    }
  }
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [CreateExtension](#) na Referência de AWS CLI Comandos.

create-hosted-configuration-version

O código de exemplo a seguir mostra como usar `create-hosted-configuration-version`.

AWS CLI

Para criar uma versão de configuração hospedada

O `create-hosted-configuration-version` exemplo a seguir cria uma nova configuração no repositório de configurações AWS AppConfig hospedado. O conteúdo da configuração deve primeiro ser convertido em base64.

```
aws appconfig create-hosted-configuration-version \
```

```
--application-id "339ohji" \  
--configuration-profile-id "ur8hx2f" \  
--  
content eyAiTmFtZSI6ICJFeGFtcGx1QXBwbGljYXRpb24iLCAiSWQiOiBFFeGFtcGx1SUQsICJSYW5rIjogMyB9  
\  
--content-type "application/json" \  
configuration_version_output_file
```

Conteúdo de configuration_version_output_file:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Saída:

```
{  
  "ApplicationId": "339ohji",  
  "ConfigurationProfileId": "ur8hx2f",  
  "VersionNumber": "1",  
  "ContentType": "application/json"  
}
```

Para obter mais informações, consulte [Sobre o armazenamento de configuração AWS AppConfig hospedado no Guia](#) do usuário do AWS AppConfig.

- Para API obter detalhes, consulte [CreateHostedConfigurationVersion](#) na Referência de AWS CLI Comandos.

delete-application

O código de exemplo a seguir mostra como usar delete-application.

AWS CLI

Como excluir uma aplicação

O delete-application exemplo a seguir exclui o aplicativo especificado.

```
aws appconfig delete-application \  
--application-id 339ohji
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteApplication](#) na Referência de AWS CLI Comandos.

delete-configuration-profile

O código de exemplo a seguir mostra como usar `delete-configuration-profile`.

AWS CLI

Para excluir um perfil de configuração

O `delete-configuration-profile` exemplo a seguir exclui o perfil de configuração especificado.

```
aws appconfig delete-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteConfigurationProfile](#) na Referência de AWS CLI Comandos.

delete-deployment-strategy

O código de exemplo a seguir mostra como usar `delete-deployment-strategy`.

AWS CLI

Para excluir uma estratégia de implantação

O `delete-deployment-strategy` exemplo a seguir exclui a estratégia de implantação especificada.

```
aws appconfig delete-deployment-strategy \  
  --deployment-strategy-id 1225qzk
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 4: Criação de uma estratégia de implantação](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteDeploymentStrategy](#) na Referência de AWS CLI Comandos.

delete-environment

O código de exemplo a seguir mostra como usar `delete-environment`.

AWS CLI

Para excluir um ambiente

O `delete-environment` exemplo a seguir exclui o ambiente de aplicativo especificado.

```
aws appconfig delete-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 2: Criação de um ambiente](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteEnvironment](#) na Referência de AWS CLI Comandos.

delete-extension-association

O código de exemplo a seguir mostra como usar `delete-extension-association`.

AWS CLI

Para excluir uma associação de extensão

O `delete-extension-association` exemplo a seguir exclui uma associação de extensão de AWS AppConfig.

```
aws appconfig delete-extension-association \  
  --region us-west-2 \  
  --extension-id 54j1r29 \  
  --extension-name 339ohji
```

```
--extension-association-id a1b2c3d4
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteExtensionAssociation](#) na Referência de AWS CLI Comandos.

delete-extension

O código de exemplo a seguir mostra como usar delete-extension.

AWS CLI

Para excluir uma extensão

O delete-extension exemplo a seguir exclui uma extensão do AWS AppConfig.

```
aws appconfig delete-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteExtension](#) na Referência de AWS CLI Comandos.

delete-hosted-configuration-version

O código de exemplo a seguir mostra como usar delete-hosted-configuration-version.

AWS CLI

Para excluir uma versão de configuração hospedada

O delete-hosted-configuration-version exemplo a seguir exclui uma versão de configuração hospedada no repositório de configuração AWS AppConfig hospedado.

```
aws appconfig delete-hosted-configuration-version \  
  --configuration-name example \  
  --extension-name example \  
  --version-id example
```

```
--application-id 339ohji \  
--configuration-profile-id ur8hx2f \  
--version-number 1
```

Saída: Esse comando não produz saída.

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [DeleteHostedConfigurationVersion](#) na Referência de AWS CLI Comandos.

get-application

O código de exemplo a seguir mostra como usar `get-application`.

AWS CLI

Para listar detalhes de um aplicativo

O `get-application` exemplo a seguir lista os detalhes do aplicativo especificado.

```
aws appconfig get-application \  
--application-id 339ohji
```

Saída:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

Para obter mais informações, consulte [Como AWS AppConfig funciona](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetApplication](#) na Referência de AWS CLI Comandos.

get-configuration-profile

O código de exemplo a seguir mostra como usar `get-configuration-profile`.

AWS CLI

Para recuperar detalhes do perfil de configuração

O `get-configuration-profile` exemplo a seguir retorna os detalhes do perfil de configuração especificado.

```
aws appconfig get-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f
```

Saída:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "ur8hx2f",  
  "Name": "Example-Configuration-Profile",  
  "LocationUri": "ssm-parameter://Example-Parameter",  
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"  
}
```

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetConfigurationProfile](#) na Referência de AWS CLI Comandos.

get-configuration

O código de exemplo a seguir mostra como usar `get-configuration`.

AWS CLI

Para recuperar detalhes da configuração

O `get-configuration` exemplo a seguir retorna os detalhes de configuração do aplicativo de exemplo. Nas chamadas subsequentes para `get-configuration`, use o `client-configuration-version` parâmetro para atualizar somente a configuração do seu aplicativo se a versão tiver sido alterada. Somente atualizar a configuração quando a versão é alterada evita cobranças excessivas incorridas ao chamar `get-configuration`.

```
aws appconfig get-configuration \  
  --application-id 339ohji \  
  --client-configuration-version 1
```



```
--application "example-application" \  
--environment "Example-Environment" \  
--configuration "Example-Configuration-Profile" \  
--client-id "test-id" \  
configuration-output-file
```

Conteúdo de configuration-output-file:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Saída:

```
{  
  "ConfigurationVersion": "1",  
  "ContentType": "application/json"  
}
```

Para obter mais informações, consulte [Etapa 6: Recebendo a configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetConfiguration](#) na Referência de AWS CLI Comandos.

get-deployment-strategy

O código de exemplo a seguir mostra como usar get-deployment-strategy.

AWS CLI

Para recuperar detalhes de uma estratégia de implantação

O get-deployment-strategy exemplo a seguir lista os detalhes da estratégia de implantação especificada.

```
aws appconfig get-deployment-strategy \  
--deployment-strategy-id 1225qzk
```

Saída:

```
{  
  "Id": "1225qzk",  
  "Name": "Example-Deployment",  
  "DeploymentDurationInMinutes": 15,
```

```
"GrowthType": "LINEAR",
"GrowthFactor": 25.0,
"FinalBakeTimeInMinutes": 0,
"ReplicateTo": "SSM_DOCUMENT"
}
```

Para obter mais informações, consulte [Etapa 4: Criação de uma estratégia de implantação](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetDeploymentStrategy](#) na Referência de AWS CLI Comandos.

get-deployment

O código de exemplo a seguir mostra como usar get-deployment.

AWS CLI

Para recuperar detalhes da implantação

O get-deployment exemplo a seguir lista os detalhes da implantação no aplicativo no ambiente e na implantação especificados.

```
aws appconfig get-deployment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --deployment-number 1
```

Saída:

```
{
  "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"  
}
```

Para obter mais informações, consulte [Etapa 5: Implantação de uma configuração](#) no Guia do AWS AppConfig usuário.

- Para API obter detalhes, consulte [GetDeployment](#) na Referência de AWS CLI Comandos.

get-environment

O código de exemplo a seguir mostra como usar `get-environment`.

AWS CLI

Para recuperar detalhes do ambiente

O `get-environment` exemplo a seguir retorna os detalhes e o estado do ambiente especificado.

```
aws appconfig get-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Saída:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "54j1r29",  
  "Name": "Example-Environment",  
  "State": "ReadyForDeployment"  
}
```

Para obter mais informações, consulte [Etapa 2: Criação de um ambiente](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetEnvironment](#) na Referência de AWS CLI Comandos.

get-extension-association

O código de exemplo a seguir mostra como usar `get-extension-association`.

AWS CLI

Para obter detalhes da associação de extensão

O `get-extension-association` exemplo a seguir exibe informações sobre uma associação de extensão.

```
aws appconfig get-extension-association \
  --region us-west-2 \
  --extension-association-id a1b2c3d4
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetExtensionAssociation](#) na Referência de AWS CLI Comandos.

get-extension

O código de exemplo a seguir mostra como usar `get-extension`.

AWS CLI

Para obter detalhes da extensão

O `get-extension` exemplo a seguir exibe informações sobre uma extensão.

```
aws appconfig get-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetExtension](#) na Referência de AWS CLI Comandos.

get-hosted-configuration-version

O código de exemplo a seguir mostra como usar `get-hosted-configuration-version`.

AWS CLI

Para recuperar detalhes da configuração hospedada

O `get-hosted-configuration-version` exemplo a seguir recupera os detalhes da configuração AWS AppConfig hospedada.

```
aws appconfig get-hosted-configuration-version \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f \  
  --version-number 1 \  
  hosted-configuration-version-output
```

Conteúdo de `hosted-configuration-version-output`:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Saída:

```
{  
  "ApplicationId": "339ohji",  
  "ConfigurationProfileId": "ur8hx2f",  
  "VersionNumber": "1",  
  "ContentType": "application/json"  
}
```

Para obter mais informações, consulte [Sobre o armazenamento de configuração AWS AppConfig hospedado](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [GetHostedConfigurationVersion](#) na Referência de AWS CLI Comandos.

list-applications

O código de exemplo a seguir mostra como usar `list-applications`.

AWS CLI

Para listar os aplicativos disponíveis

O `list-applications` exemplo a seguir lista os aplicativos disponíveis em sua AWS conta.

```
aws appconfig list-applications
```

Saída:

```
{
  "Items": [
    {
      "Id": "339ohji",
      "Name": "test-application",
      "Description": "An application used for creating an example."
    },
    {
      "Id": "rwalwu7",
      "Name": "Test-Application"
    }
  ]
}
```

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListApplications](#) na Referência de AWS CLI Comandos.

list-configuration-profiles

O código de exemplo a seguir mostra como usar `list-configuration-profiles`.

AWS CLI

Para listar os perfis de configuração disponíveis

O `list-configuration-profiles` exemplo a seguir lista os perfis de configuração disponíveis para o aplicativo especificado.

```
aws appconfig list-configuration-profiles \
  --application-id 339ohji
```

Saída:

```
{
  "Items": [
    {
      "ApplicationId": "339ohji",
      "Id": "ur8hx2f",
      "Name": "Example-Configuration-Profile",
      "LocationUri": "ssm-parameter://Example-Parameter"
    }
  ]
}
```



```

    }
  ]
}

```

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListConfigurationProfiles](#) na Referência de AWS CLI Comandos.

list-deployment-strategies

O código de exemplo a seguir mostra como usar `list-deployment-strategies`.

AWS CLI

Para listar as estratégias de implantação disponíveis

O `list-deployment-strategies` exemplo a seguir lista as estratégias de implantação disponíveis em sua AWS conta.

```
aws appconfig list-deployment-strategies
```

Saída:

```

{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Etapa 4: Criação de uma estratégia de implantação](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListDeploymentStrategies](#) na Referência de AWS CLI Comandos.

list-deployments

O código de exemplo a seguir mostra como usar `list-deployments`.

AWS CLI

Para listar as implantações disponíveis

O `list-deployments` exemplo a seguir lista as implantações disponíveis em sua AWS conta para o aplicativo e o ambiente especificados.

```
aws appconfig list-deployments \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Etapa 5: Implantação de uma configuração](#) no Guia do AWS AppConfig usuário.

- Para API obter detalhes, consulte [ListDeployments](#) na Referência de AWS CLI Comandos.

list-environments

O código de exemplo a seguir mostra como usar `list-environments`.

AWS CLI

Para listar os ambientes disponíveis

O `list-environments` exemplo a seguir lista os ambientes disponíveis em sua AWS conta para o aplicativo especificado.

```
aws appconfig list-environments \  
  --application-id 339ohji
```

Saída:

```
{
  "Items": [
    {
      "ApplicationId": "339ohji",
      "Id": "54j1r29",
      "Name": "Example-Environment",
      "State": "ReadyForDeployment"
    }
  ]
}
```

Para obter mais informações, consulte [Etapa 2: Criação de um ambiente](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListEnvironments](#) na Referência de AWS CLI Comandos.

list-extension-associations

O código de exemplo a seguir mostra como usar `list-extension-associations`.

AWS CLI

Para listar todas as associações de AWS AppConfig extensão em sua AWS conta para uma AWS região

O `list-extension-associations` exemplo a seguir lista todas as associações de AWS AppConfig extensão da AWS conta atual em uma AWS região específica.

```
aws appconfig list-extension-associations \
  --region us-west-2
```

Saída:

```
{
  "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"  
  }  
]  
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListExtensionAssociations](#) na Referência de AWS CLI Comandos.

list-extensions

O código de exemplo a seguir mostra como usar `list-extensions`.

AWS CLI

Para listar todas AWS AppConfig as extensões em sua AWS conta para uma AWS região

O `list-extensions` exemplo a seguir lista todas as AWS AppConfig extensões da AWS conta atual em uma AWS região específica. O comando retorna extensões personalizadas e criadas por AWS autoria.

```
aws appconfig list-extensions \  
  --region us-west-2
```

Saída:

```
{  
  "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."
    }
]
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListExtensions](#) na Referência de AWS CLI Comandos.

list-hosted-configuration-versions

O código de exemplo a seguir mostra como usar `list-hosted-configuration-versions`.

AWS CLI

Para listar as versões de configuração hospedada disponíveis

O `list-hosted-configuration-versions` exemplo a seguir lista as versões de configuração hospedadas no repositório de configurações AWS AppConfig hospedado para o aplicativo e o perfil de configuração especificados.

```
aws appconfig list-hosted-configuration-versions \
  --application-id 339ohji \
  --configuration-profile-id ur8hx2f
```

Saída:

```
{
  "Items": [
    {
      "ApplicationId": "339ohji",
      "ConfigurationProfileId": "ur8hx2f",
      "VersionNumber": 1,
      "ContentType": "application/json"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Sobre o armazenamento de configuração AWS AppConfig hospedado](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListHostedConfigurationVersions](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um aplicativo

O `list-tags-for-resource` exemplo a seguir lista as tags de um aplicativo especificado.

```
aws appconfig list-tags-for-resource \
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

Saída:

```
{
  "Tags": {
    "group1": "1"
  }
}
```

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

start-deployment

O código de exemplo a seguir mostra como usar `start-deployment`.

AWS CLI

Para iniciar uma implantação de configuração

O `start-deployment` exemplo a seguir inicia uma implantação no aplicativo usando o ambiente, a estratégia de implantação e o perfil de configuração especificados.

```
aws appconfig start-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-strategy-id 1225qzk \  
  --configuration-profile-id ur8hx2f \  
  --configuration-version 1
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Etapa 5: Implantação de uma configuração](#) no Guia do AWS AppConfig usuário.

- Para API obter detalhes, consulte [StartDeployment](#) na Referência de AWS CLI Comandos.

stop-deployment

O código de exemplo a seguir mostra como usar stop-deployment.

AWS CLI

Para interromper a implantação da configuração

O stop-deployment exemplo a seguir interrompe a implantação de uma configuração de aplicativo no ambiente especificado.

```
aws appconfig stop-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-number 2
```

Saída:

```
{  
  "DeploymentNumber": 0,  
  "DeploymentDurationInMinutes": 0,  
  "GrowthFactor": 0.0,  
  "FinalBakeTimeInMinutes": 0,  
  "PercentageComplete": 0.0  
}
```

Para obter mais informações, consulte [Etapa 5: Implantação de uma configuração](#) no Guia do AWS AppConfig usuário.

- Para API obter detalhes, consulte [StopDeployment](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar um aplicativo

O tag-resource exemplo a seguir marca um recurso do aplicativo.

```
aws appconfig tag-resource \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-number 2
```

```
--resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \  
--tags '{"group1" : "1"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover uma tag de um aplicativo

O untag-resource exemplo a seguir remove a tag group1 do aplicativo especificado.

```
aws appconfig untag-resource \  
--resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \  
--tag-keys '["group1"]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-application

O código de exemplo a seguir mostra como usar update-application.

AWS CLI

Para atualizar um aplicativo

O update-application exemplo a seguir atualiza o nome do aplicativo especificado.

```
aws appconfig update-application \  
--application-id 339ohji \  
--application-name 339ohji
```

```
--name "Example-Application"
```

Saída:

```
{
  "Id": "339ohji",
  "Name": "Example-Application",
  "Description": "An application used for creating an example."
}
```

Para obter mais informações, consulte [Etapa 1: Criação de um AWS AppConfig aplicativo](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UpdateApplication](#) na Referência de AWS CLI Comandos.

update-configuration-profile

O código de exemplo a seguir mostra como usar update-configuration-profile.

AWS CLI

Para atualizar um perfil de configuração

O update-configuration-profile exemplo a seguir atualiza a descrição do perfil de configuração especificado.

```
aws appconfig update-configuration-profile \
  --application-id 339ohji \
  --configuration-profile-id ur8hx2f \
  --description "Configuration profile used for examples."
```

Saída:

```
{
  "ApplicationId": "339ohji",
  "Id": "ur8hx2f",
  "Name": "Example-Configuration-Profile",
  "Description": "Configuration profile used for examples.",
  "LocationUri": "ssm-parameter://Example-Parameter",
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UpdateConfigurationProfile](#) na Referência de AWS CLI Comandos.

update-deployment-strategy

O código de exemplo a seguir mostra como usar `update-deployment-strategy`.

AWS CLI

Para atualizar uma estratégia de implantação

O `update-deployment-strategy` exemplo a seguir atualiza o tempo final de cozimento para 20 minutos na estratégia de implantação especificada.

```
aws appconfig update-deployment-strategy \  
  --deployment-strategy-id 1225qzk \  
  --final-bake-time-in-minutes 20
```

Saída:

```
{  
  "Id": "1225qzk",  
  "Name": "Example-Deployment",  
  "DeploymentDurationInMinutes": 15,  
  "GrowthType": "LINEAR",  
  "GrowthFactor": 25.0,  
  "FinalBakeTimeInMinutes": 20,  
  "ReplicateTo": "SSM_DOCUMENT"  
}
```

Para obter mais informações, consulte [Etapa 4: Criação de uma estratégia de implantação](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UpdateDeploymentStrategy](#) na Referência de AWS CLI Comandos.

update-environment

O código de exemplo a seguir mostra como usar `update-environment`.

AWS CLI

Para atualizar um ambiente

O `update-environment` exemplo a seguir atualiza a descrição de um ambiente.

```
aws appconfig update-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --description "An environment for examples."
```

Saída:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "54j1r29",  
  "Name": "Example-Environment",  
  "Description": "An environment for examples.",  
  "State": "RolledBack"  
}
```

Para obter mais informações, consulte [Etapa 2: Criação de um ambiente](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UpdateEnvironment](#) na Referência de AWS CLI Comandos.

update-extension-association

O código de exemplo a seguir mostra como usar `update-extension-association`.

AWS CLI

Para atualizar uma associação AWS AppConfig de extensão

O `update-extension-association` exemplo a seguir adiciona um novo valor de parâmetro a uma associação de extensão em AWS AppConfig.

```
aws appconfig update-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4 \  
  --parameters S3bucket=FinanceMobileApp
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UpdateExtensionAssociation](#) na Referência de AWS CLI Comandos.

update-extension

O código de exemplo a seguir mostra como usar update-extension.

AWS CLI

Para atualizar uma AWS AppConfig extensão

O update-extension exemplo a seguir adiciona um parâmetro adicional Key a uma extensão em AWS AppConfig.

```
aws appconfig update-extension \
  --region us-west-2 \
  --extension-identifier S3-backup-extension \
  --parameters S3bucket={Required=true}, CampaignID={Required=false}
```

Saída:

```
{
  "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
  }
}
}

```

Para obter mais informações, consulte Como [trabalhar com AWS AppConfig extensões](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [UpdateExtension](#) na Referência de AWS CLI Comandos.

validate-configuration

O código de exemplo a seguir mostra como usar `validate-configuration`.

AWS CLI

Para validar uma configuração

O `validate-configuration` exemplo a seguir usa os validadores em um perfil de configuração para validar uma configuração.

```

aws appconfig validate-configuration \
  --application-id abc1234 \
  --configuration-profile-id ur8hx2f \
  --configuration-version 1

```

O comando não produz nenhuma saída.

Para obter mais informações, consulte [Etapa 3: Criar uma configuração e um perfil de configuração](#) no Guia AWS AppConfig do usuário.

- Para API obter detalhes, consulte [ValidateConfiguration](#) na Referência de AWS CLI Comandos.

Exemplos de Application Auto Scaling usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Application Auto Scaling.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-scaling-policy

O código de exemplo a seguir mostra como usar delete-scaling-policy.

AWS CLI

Para excluir uma política de escalabilidade

Este exemplo exclui uma política de escalabilidade para o aplicativo web de ECS serviços da Amazon em execução no cluster padrão.

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
```

- Para API obter detalhes, consulte [DeleteScalingPolicy](#) na Referência de AWS CLI Comandos.

delete-scheduled-action

O código de exemplo a seguir mostra como usar `delete-scheduled-action`.

AWS CLI

Para excluir uma ação programada

O `delete-scheduled-action` exemplo a seguir exclui a ação programada especificada da frota Amazon AppStream 2.0 especificada:

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Escalabilidade programada](#) no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [DeleteScheduledAction](#) na Referência de AWS CLI Comandos.

deregister-scalable-target

O código de exemplo a seguir mostra como usar `deregister-scalable-target`.

AWS CLI

Para cancelar o registro de uma meta escalável

Este exemplo cancela o registro de um destino escalável para um ECS serviço da Amazon chamado `web-app` que está sendo executado no cluster padrão.

Comando:

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --  
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

Este exemplo cancela o registro de uma meta escalável para um recurso personalizado. O `custom-resource-id` arquivo.txt contém uma string que identifica o ID do recurso, que, para um

recurso personalizado, é o caminho para o recurso personalizado por meio do endpoint do 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
```

Conteúdo do custom-resource-id arquivo.txt:

```
https://example.execute-api.us-west-2.amazonaws.com/prod/scalableTargetDimensions/1-23456789
```

- Para API obter detalhes, consulte [DeregisterScalableTarget](#) na Referência de AWS CLI Comandos.

describe-scalable-targets

O código de exemplo a seguir mostra como usar describe-scalable-targets.

AWS CLI

Para descrever metas escaláveis

O describe-scalable-targets exemplo a seguir descreve os destinos escaláveis para o namespace do ecs serviço.

```
aws application-autoscaling describe-scalable-targets \  
--service-namespace ecs
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [AWS os serviços que você pode usar com o Application Auto Scaling](#) no Guia do Usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [DescribeScalableTargets](#) na Referência de AWS CLI Comandos.

describe-scaling-activities

O código de exemplo a seguir mostra como usar `describe-scaling-activities`.

AWS CLI

Exemplo 1: Para descrever as atividades de escalabilidade para o serviço Amazon ECS especificado

O `describe-scaling-activities` exemplo a seguir descreve as atividades de escalabilidade de um ECS serviço da Amazon chamado `web-app` que está sendo executado no `default` cluster. A saída mostra uma atividade de escalabilidade iniciada por uma política de escalabilidade.

```

aws application-autoscaling describe-scaling-activities \
  --service-namespace ecs \
  --resource-id service/default/web-app

```

Saída:

```
{
```

```

"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"
  }
]
}

```

Para obter mais informações, consulte [Atividades de escalabilidade para Application Auto Scaling](#) no Guia do usuário do Application Auto Scaling.

Exemplo 2: Para descrever as atividades de escalabilidade para a tabela especificada do DynamoDB

O `describe-scaling-activities` exemplo a seguir descreve as atividades de escalabilidade de uma tabela do DynamoDB chamada. `TestTable` O resultado mostra as atividades de escalabilidade iniciadas por duas ações agendadas diferentes.

```

aws application-autoscaling describe-scaling-activities \
  --service-namespace dynamodb \
  --resource-id table/TestTable

```

Saída:

```

{
  "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"
      }
    ]
  }
```

Para obter mais informações, consulte [Atividades de escalabilidade para Application Auto Scaling](#) no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [DescribeScalingActivities](#) na Referência de AWS CLI Comandos.

describe-scaling-policies

O código de exemplo a seguir mostra como usar `describe-scaling-policies`.

AWS CLI

Para descrever as políticas de escalabilidade

Este exemplo de comando descreve as políticas de escalabilidade para o namespace do serviço `ecs`.

Comando:

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

Saída:

```
{
  "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-lt-25",
    "PolicyType": "StepScaling",
    "Alarms": [
        {
            "AlarmName": "web-app-cpu-lt-25",
            "AlarmARN": "arn:aws:cloudwatch:us-
west-2:012345678910:alarm:web-app-cpu-lt-25"
        }
    ],
    "ServiceNamespace": "ecs"
}
]
```



```
}
```

- Para API obter detalhes, consulte [DescribeScalingPolicies](#) na Referência de AWS CLI Comandos.

describe-scheduled-actions

O código de exemplo a seguir mostra como usar `describe-scheduled-actions`.

AWS CLI

Para descrever ações agendadas

O `describe-scheduled-actions` exemplo a seguir exibe detalhes das ações agendadas para o namespace de serviço especificado:

```
aws application-autoscaling describe-scheduled-actions \  
  --service-namespace dynamodb
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Escalabilidade programada](#) no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [DescribeScheduledActions](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um alvo escalável

O `list-tags-for-resource` exemplo a seguir lista os nomes e valores das chaves de tag que estão anexados ao alvo escalável especificado por itsARN.

```

aws application-autoscaling list-tags-for-resource \
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-
target/1234abcd56ab78cd901ef1234567890ab123

```

Saída:

```

{
  "Tags": {
    "environment": "production"
  }
}

```

```
}
}
```

Para obter mais informações, consulte [Suporte de marcação para Application Auto Scaling](#) no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-scaling-policy

O código de exemplo a seguir mostra como usar `put-scaling-policy`.

AWS CLI

Exemplo 1: como aplicar uma política de escalabilidade com monitoramento do objetivo com uma especificação de métrica predefinida

O `put-scaling-policy` exemplo a seguir aplica uma política de escalabilidade de rastreamento de metas com uma especificação métrica predefinida a um ECS serviço da Amazon chamado `web-app` no cluster padrão. A política mantém a CPU utilização média do serviço em 75 por cento, com períodos de espera de expansão e expansão de 60 segundos. A saída contém os nomes ARNs e dos dois CloudWatch alarmes criados em seu nome.

```
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
```

Este exemplo pressupõe que você tenha um arquivo `config.json` no diretório atual com o seguinte conteúdo:

```
{
  "TargetValue": 75.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
  },
  "ScaleOutCooldown": 60,
  "ScaleInCooldown": 60
}
```

```
}

```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: como aplicar uma política de escalabilidade com monitoramento do objetivo com uma especificação de métrica personalizada

O `put-scaling-policy` exemplo a seguir aplica uma política de escalabilidade de rastreamento de metas com uma especificação métrica personalizada a um ECS serviço da Amazon chamado `web-app` no cluster padrão. A política mantém a utilização média do serviço em 75 por cento, com períodos de espera de expansão e expansão de 60 segundos. A saída contém os nomes ARNs e dos dois CloudWatch alarmes criados em seu nome.

```
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
```

Este exemplo pressupõe que você tenha um arquivo `config.json` no diretório atual com o seguinte conteúdo:

```
{
  "TargetValue":75.0,
  "CustomizedMetricSpecification":{
    "MetricName":"MyUtilizationMetric",
    "Namespace":"MyNamespace",
    "Dimensions": [
      {
        "Name":"MyOptionalMetricDimensionName",
        "Value":"MyOptionalMetricDimensionValue"
      }
    ],
    "Statistic":"Average",
    "Unit":"Percent"
  },
  "ScaleOutCooldown": 60,
  "ScaleInCooldown": 60
}
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

Exemplo 3: como aplicar uma política de escalabilidade com monitoramento do objetivo somente para expansão

O `put-scaling-policy` exemplo a seguir aplica uma política de escalabilidade de rastreamento de metas a um ECS serviço da Amazon chamado `web-app` no cluster padrão. A política é usada para escalar o ECS serviço quando a `RequestCountPerTarget` métrica do Application Load Balancer excede o limite. A saída contém o nome ARN e o nome do CloudWatch alarme criado em seu nome.

```
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
```

Conteúdo de `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
}
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Políticas de escalabilidade de rastreamento de metas para Application Auto Scaling](#) no Guia do usuário AWS do Application Auto Scaling.

- Para API obter detalhes, consulte [PutScalingPolicy](#) na Referência de AWS CLI Comandos.

put-scheduled-action

O código de exemplo a seguir mostra como usar `put-scheduled-action`.

AWS CLI

Para adicionar uma ação agendada a uma tabela do DynamoDB

Este exemplo adiciona uma ação programada a uma tabela do DynamoDB `TestTable` chamada para escalar de acordo com uma programação recorrente. Na programação especificada (todos os dias às 12h15UTC), se a capacidade atual estiver abaixo do valor especificado para `MinCapacity`, o Application Auto Scaling se expande até o valor especificado por `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

```

Para obter mais informações, consulte Scheduled Scaling no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [PutScheduledAction](#) na Referência de AWS CLI Comandos.

register-scalable-target

O código de exemplo a seguir mostra como usar `register-scalable-target`.

AWS CLI

Exemplo 1: Para registrar um ECS serviço como um alvo escalável

O `register-scalable-target` exemplo a seguir registra um ECS serviço da Amazon com o Application Auto Scaling. Ele também adiciona uma tag com o nome da chave `environment` e o valor `production` ao alvo escalável.

```
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
```

Saída:

```
{  
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-  
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"  
}
```

Para obter exemplos de outros AWS serviços e recursos personalizados, consulte os tópicos em [AWS serviços que você pode usar com o Application Auto Scaling](#) no Guia do Usuário do Application Auto Scaling.

Exemplo 2: Para suspender as atividades de escalabilidade para um alvo escalável

O `register-scalable-target` exemplo a seguir suspende as atividades de escalabilidade de um alvo escalável existente.

```
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,ScheduledScalingSuspenda
```

Saída:

```
{
```



```
"ScalableTargetARN": "arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Para obter mais informações, consulte [Suspende e retomar o escalonamento do Application Auto Scaling no Guia do usuário do Application Auto Scaling](#).

Exemplo 3: Para retomar as atividades de escalabilidade para uma meta escalável

O `register-scalable-target` exemplo a seguir retoma as atividades de escalabilidade para uma meta escalável existente.

```
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
```

Saída:

```
{
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Para obter mais informações, consulte [Suspende e retomar o escalonamento do Application Auto Scaling no Guia do usuário do Application Auto Scaling](#).

- Para API obter detalhes, consulte [RegisterScalableTarget](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a um alvo escalável

O `tag-resource` exemplo a seguir adiciona uma tag com o nome da chave `environment` e o valor `production` ao alvo escalável especificado por `itsARN`.

```
aws application-autoscaling tag-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123 \  
  --tags environment=production
```

Este comando não produz saída.

Para obter mais informações, consulte [Suporte de marcação para Application Auto Scaling](#) no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover uma tag de um alvo escalável

O untag-resource exemplo a seguir remove o par de tags com o nome `environment` da chave do destino escalável especificado por itsARN.

```
aws application-autoscaling untag-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123 \  
  --tag-keys "environment"
```

Este comando não produz saída.

Para obter mais informações, consulte [Suporte de marcação para Application Auto Scaling](#) no Guia do usuário do Application Auto Scaling.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

Exemplos do Application Discovery Service usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Application Discovery Service.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-agents

O código de exemplo a seguir mostra como usar `describe-agents`.

AWS CLI

Descreva agentes com `collectionStatus` estados especificados

Este exemplo de comando descreve agentes de coleta com status de coleção de `STARTED` ou `STOPPED`.

Comando:

```
aws discovery describe-agents --filters
  name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
  results 3
```

Saída:

```
{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [DescribeAgents](#) na Referência de AWS CLI Comandos.

describe-configurations

O código de exemplo a seguir mostra como usar `describe-configurations`.

AWS CLI

Descrever as configurações de ativos selecionadas

Este exemplo de comando descreve as configurações de dois servidores especificados. A ação detecta o tipo de ativo a partir do ID de configuração. Somente um tipo de ativo é permitido por comando.

Comando:

```
aws discovery describe-configurations --configuration-ids "d-
server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

Saída:

```
{
  "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"
    }
]
}

```

Descrever as configurações de ativos selecionadas

Este exemplo de comando descreve as configurações de dois aplicativos especificados. A ação detecta o tipo de ativo a partir do ID de configuração. Somente um tipo de ativo é permitido por comando.

Comando:

```

aws discovery describe-configurations --configuration-ids "d-
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"

```

Saída:

```

{
  "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"
      }
    ]
  }

```

- Para API obter detalhes, consulte [DescribeConfigurations](#) na Referência de AWS CLI Comandos.

list-configurations

O código de exemplo a seguir mostra como usar `list-configurations`.

AWS CLI

Para listar todos os servidores descobertos que atendem a um conjunto de condições de filtro

Este exemplo de comando lista os servidores descobertos que correspondem a um dos dois padrões de nome de host e que não estão executando o 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"

```

Saída:

```

{
  "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-099385097ef9fbcbf",
      "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"
  }
]
}
```

- Para API obter detalhes, consulte [ListConfigurations](#) na Referência de AWS CLI Comandos.

AppRegistry exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AppRegistry.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-attribute-group

O código de exemplo a seguir mostra como usar `associate-attribute-group`.

AWS CLI

Para associar um grupo de atributos

O `associate-attribute-group` exemplo a seguir associa um grupo de atributos específico em sua AWS conta a um aplicativo específico em sua AWS conta.

```
aws servicecatalog-appregistry associate-attribute-group \
```

```
--application "ExampleApplication" \  
--attribute-group "ExampleAttributeGroup"
```

Saída:

```
{  
  "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
  "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1"  
}
```

Para obter mais informações, consulte [Associando e desassociando grupos de atributos](#) no AWS Service Catalog Administrator Guide. AppRegistry

- Para API obter detalhes, consulte [AssociateAttributeGroup](#) na Referência de AWS CLI Comandos.

create-application

O código de exemplo a seguir mostra como usar create-application.

AWS CLI

Para criar um aplicativo

O create-application exemplo a seguir cria um novo aplicativo em sua AWS conta.

```
aws servicecatalog-appregistry create-application \  
--name "ExampleApplication"
```

Saída:

```
{  
  "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": {}
  }
}

```

Para obter mais informações, consulte [Criação de aplicativos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [CreateApplication](#) na Referência de AWS CLI Comandos.

create-attribute-group

O código de exemplo a seguir mostra como usar create-attribute-group.

AWS CLI

Para criar um grupo de atributos

O create-attribute-group exemplo a seguir cria um novo grupo de atributos na sua AWS conta.

```

aws servicecatalog-appregistry create-attribute-group \
  --name "ExampleAttributeGroup" \
  --attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'

```

Saída:

```

{
  "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": {}
  }
}

```

Para obter mais informações, consulte [Criação de grupos de atributos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [CreateAttributeGroup](#) na Referência de AWS CLI Comandos.

delete-application

O código de exemplo a seguir mostra como usar delete-application.

AWS CLI

Como excluir uma aplicação

O delete-application exemplo a seguir exclui um aplicativo específico da sua AWS conta.

```
aws servicecatalog-appregistry delete-application \  
  --application "ExampleApplication3"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Excluindo aplicativos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [DeleteApplication](#) na Referência de AWS CLI Comandos.

delete-attribute-group

O código de exemplo a seguir mostra como usar delete-attribute-group.

AWS CLI

Exemplo 8: Para excluir um grupo de atributos

O `delete-attribute-group` exemplo a seguir exclui um grupo de atributos específico em sua AWS conta.

```
aws servicecatalog-appregistry delete-attribute-group \  
--attribute-group "ExampleAttributeGroup3"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Excluindo grupos de atributos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [DeleteAttributeGroup](#) na Referência de AWS CLI Comandos.

get-application

O código de exemplo a seguir mostra como usar `get-application`.

AWS CLI

Para obter um aplicativo

O `get-application` exemplo a seguir recupera informações de metadados sobre um aplicativo específico em sua AWS conta.

```
aws servicecatalog-appregistry get-application \  
--application "ExampleApplication"
```

Saída:

```
{
```

```

    "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"
      }
    }
  }
}

```

Para obter mais informações, consulte [Usando detalhes do aplicativo](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [GetApplication](#) na Referência de AWS CLI Comandos.

get-attribute-group

O código de exemplo a seguir mostra como usar get-attribute-group.

AWS CLI

Para obter um grupo de atributos

O get-attribute-group exemplo a seguir recupera um grupo de atributos específico em sua AWS conta.

```

aws servicecatalog-appregistry get-attribute-group \
  --attribute-group ExampleAttributeGroup

```

Saída:

```

{
  "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"
}
}
```

Para obter mais informações, consulte [Gerenciamento de metadados para grupos de atributos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [GetAttributeGroup](#) na Referência de AWS CLI Comandos.

list-applications

O código de exemplo a seguir mostra como usar `list-applications`.

AWS CLI

Para listar aplicativos

O `list-applications` exemplo a seguir recupera uma lista de todos os aplicativos em sua AWS conta.

```
aws servicecatalog-appregistry list-applications
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Visualizando detalhes do aplicativo](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [ListApplications](#) na Referência de AWS CLI Comandos.

list-associated-attribute-groups

O código de exemplo a seguir mostra como usar `list-associated-attribute-groups`.

AWS CLI

Para listar grupos de atributos associados

O `list-associated-attribute-groups` exemplo a seguir recupera uma lista de todos os grupos de atributos em sua AWS conta que estão associados a um aplicativo específico em sua AWS conta.

```
aws servicecatalog-appregistry list-associated-attribute-groups \
  --application "ExampleApplication"
```

Saída:

```
{
```



```
"attributeGroups": [  
    "01sj5xdwhbw54kejwnt09fnpc1"  
]  
}
```

Para obter mais informações, consulte [Associando e desassociando grupos de atributos](#) no AWS Service Catalog Administrator Guide. AppRegistry

- Para API obter detalhes, consulte [ListAssociatedAttributeGroups](#) na Referência de AWS CLI Comandos.

list-attribute-groups-for-application

O código de exemplo a seguir mostra como usar `list-attribute-groups-for-application`.

AWS CLI

Para listar grupos de atributos para um aplicativo

O `list-attribute-groups-for-application` exemplo a seguir lista os detalhes de todos os grupos de atributos em sua AWS conta que estão associados a um aplicativo específico em sua AWS conta.

```
aws servicecatalog-appregistry list-attribute-groups-for-application \  
--application "ExampleApplication"
```

Saída:

```
{  
  "attributeGroupsDetails": [  
    {  
      "id": "01sj5xdwhbw54kejwnt09fnpc1",  
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
      "name": "ExampleAttributeGroup"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Visualizando detalhes do grupo de atributos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [ListAttributeGroupsForApplication](#) na Referência de AWS CLI Comandos.

list-attribute-groups

O código de exemplo a seguir mostra como usar `list-attribute-groups`.

AWS CLI

Para listar grupos de atributos

O `list-attribute-groups` exemplo a seguir recupera uma lista de todos os grupos de atributos em sua AWS conta.

```
aws servicecatalog-appregistry list-attribute-groups
```

Saída:

```
{
  "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"
    }
]
}
```

Para obter mais informações, consulte [Visualizando detalhes do grupo de atributos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [ListAttributeGroups](#) na Referência de AWS CLI Comandos.

update-application

O código de exemplo a seguir mostra como usar `update-application`.

AWS CLI

Para atualizar um aplicativo

O `update-application` exemplo a seguir atualiza um aplicativo específico em sua AWS conta para incluir uma descrição.

```
aws servicecatalog-appregistry update-application \
  --application "ExampleApplication" \
  --description "This is an example application"
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Editando aplicativos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [UpdateApplication](#) na Referência de AWS CLI Comandos.

update-attribute-group

O código de exemplo a seguir mostra como usar update-attribute-group.

AWS CLI

Para atualizar um grupo de atributos

O update-attribute-group exemplo a seguir atualiza um grupo de atributos específico em sua AWS conta para incluir uma descrição.

```
aws servicecatalog-appregistry update-attribute-group \  
  --attribute-group "ExampleAttributeGroup" \  
  --description "This is an example attribute group"
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Editando grupos de atributos](#) no AWS Service Catalog AppRegistry Administrator Guide.

- Para API obter detalhes, consulte [UpdateAttributeGroup](#) na Referência de AWS CLI Comandos.

Exemplos de Athena usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Athena.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-get-named-query

O código de exemplo a seguir mostra como usar `batch-get-named-query`.

AWS CLI

Para retornar informações sobre mais de uma consulta

O `batch-get-named-query` exemplo a seguir retorna informações sobre as consultas nomeadas que têm o especificadoIDs.

```
aws athena batch-get-named-query \  
  --named-query-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

Saída:

```
{  
  "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": []
}

```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [BatchGetNamedQuery](#) na Referência de AWS CLI Comandos.

batch-get-query-execution

O código de exemplo a seguir mostra como usar batch-get-query-execution.

AWS CLI

Para retornar informações sobre uma ou mais execuções de consulta

O `batch-get-query-execution` exemplo a seguir retorna informações de execução da consulta para as consultas que têm a consulta IDs especificada.

```
aws athena batch-get-query-execution \
  --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-
  cdef-EXAMPLE22222
```

Saída:

```
{
  "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": []
}

```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [BatchGetQueryExecution](#) na Referência de AWS CLI Comandos.

create-data-catalog

O código de exemplo a seguir mostra como usar create-data-catalog.

AWS CLI

Para criar um catálogo de dados

O create-data-catalog exemplo a seguir cria o catálogo dynamo_db_catalog de dados.

```

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
```

Este comando não produz saída. Para ver o resultado, use `aws athena get-data-catalog --name dynamo_db_catalog`.

Para obter mais informações, consulte [Registro de um catálogo: create-data-catalog](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [CreateDataCatalog](#) na Referência de AWS CLI Comandos.

create-named-query

O código de exemplo a seguir mostra como usar `create-named-query`.

AWS CLI

Para criar uma consulta nomeada

O `create-named-query` exemplo a seguir cria uma consulta salva no AthenaAdmin grupo de trabalho que consulta a `flights_parquet` tabela de voos de Seattle para Nova York em janeiro de 2016, cuja partida e chegada foram atrasadas em mais de dez minutos. Como os valores do código do aeroporto na tabela são cadeias de caracteres que incluem aspas duplas (por exemplo, "SEA"), eles são escapados por barras invertidas e cercados por aspas simples.

```
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
```

Saída:

```
{  
  "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [CreateNamedQuery](#) na Referência de AWS CLI Comandos.

create-work-group

O código de exemplo a seguir mostra como usar `create-work-group`.

AWS CLI

Para criar um grupo de trabalho

O `create-work-group` exemplo a seguir cria um grupo de trabalho chamado `Data_Analyst_Group` que tem o local `s3://awsdoc-example-bucket` de saída dos resultados da consulta. O comando cria um grupo de trabalho que substitui as configurações do cliente, o que inclui o local de saída dos resultados da consulta. O comando também ativa CloudWatch métricas e adiciona três pares de tags de valores-chave ao grupo de trabalho para diferenciá-lo de outros grupos de trabalho. Observe que o `--configuration` argumento não tem espaços antes das vírgulas que separam suas opções.

```
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"
```

Este comando não produz saída. Para ver os resultados, use `aws athena get-work-group --work-group Data_Analyst_Group`.

Para obter mais informações, consulte [Gerenciamento de grupos de trabalho](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [CreateWorkGroup](#) na Referência de AWS CLI Comandos.

delete-data-catalog

O código de exemplo a seguir mostra como usar `delete-data-catalog`.

AWS CLI

Para excluir um catálogo de dados

O `delete-data-catalog` exemplo a seguir exclui o catálogo `UnusedDataCatalog` de dados.

```
aws athena delete-data-catalog \  
  --name UnusedDataCatalog
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um catálogo: delete-data-catalog](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [DeleteDataCatalog](#) na Referência de AWS CLI Comandos.

delete-named-query

O código de exemplo a seguir mostra como usar `delete-named-query`.

AWS CLI

Para excluir uma consulta nomeada

O `delete-named-query` exemplo a seguir exclui a consulta nomeada que tem o ID especificado.

```
aws athena delete-named-query \  
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Este comando não produz saída.

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [DeleteNamedQuery](#) na Referência de AWS CLI Comandos.

delete-work-group

O código de exemplo a seguir mostra como usar `delete-work-group`.

AWS CLI

Para excluir um grupo de trabalho

O `delete-work-group` exemplo a seguir exclui o grupo de `TeamB` trabalho.

```
aws athena delete-work-group \  
  --work-group TeamB
```

Este comando não produz saída. Para confirmar a exclusão, use `aws athena list-work-groups`.

Para obter mais informações, consulte [Gerenciamento de grupos de trabalho](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [DeleteWorkGroup](#) na Referência de AWS CLI Comandos.

get-data-catalog

O código de exemplo a seguir mostra como usar `get-data-catalog`.

AWS CLI

Para retornar informações sobre um catálogo de dados

O `get-data-catalog` exemplo a seguir retorna informações sobre o catálogo `dynamo_db_catalog` de dados.

```
aws athena get-data-catalog \  
  --name dynamo_db_catalog
```

Saída:

```
{  
  "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"
    }
}
}

```

Para obter mais informações, consulte [Exibindo detalhes do catálogo: get-data-catalog](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetDataCatalog](#) na Referência de AWS CLI Comandos.

get-database

O código de exemplo a seguir mostra como usar get-database.

AWS CLI

Para retornar informações sobre um banco de dados em um catálogo de dados

O get-database exemplo a seguir retorna informações sobre o `sampledb` banco de dados no catálogo `AwsDataCatalog` de dados.

```

aws athena get-database \
  --catalog-name AwsDataCatalog \
  --database-name sampledb

```

Saída:

```

{
  "Database": {
    "Name": "sampledb",
    "Description": "Sample database",
    "Parameters": {
      "CreatedBy": "Athena",
      "EXTERNAL": "TRUE"
    }
  }
}

```

Para obter mais informações, consulte [Mostrando detalhes do banco de dados: get-database no Guia](#) do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetDatabases](#) na Referência de AWS CLI Comandos.

get-named-query

O código de exemplo a seguir mostra como usar `get-named-query`.

AWS CLI

Para retornar uma consulta nomeada

O `get-named-query` exemplo a seguir retorna informações sobre a consulta que tem o ID especificado.

```
aws athena get-named-query \  
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetNamedQuery](#) na Referência de AWS CLI Comandos.

get-query-execution

O código de exemplo a seguir mostra como usar `get-query-execution`.

AWS CLI

Para retornar informações sobre a execução de uma consulta

O `get-query-execution` exemplo a seguir retorna informações sobre a consulta que tem o ID de consulta especificado.

```
aws athena get-query-execution \  
--query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetQueryExecution](#) na Referência de AWS CLI Comandos.

get-query-results

O código de exemplo a seguir mostra como usar `get-query-results`.

AWS CLI

Para retornar os resultados de uma consulta

O `get-query-results` exemplo a seguir retorna os resultados da consulta que tem o ID de consulta especificado.

```
aws athena get-query-results \  
--query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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
}

```

Para obter mais informações, consulte Como [trabalhar com resultados de consultas, arquivos de saída e histórico de consultas](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetQueryResults](#) na Referência de AWS CLI Comandos.

get-table-metadata

O código de exemplo a seguir mostra como usar `get-table-metadata`.

AWS CLI

Para retornar informações de metadados sobre uma tabela

O `get-table-metadata` exemplo a seguir retorna informações de metadados sobre a `counties` tabela, incluindo nomes de colunas e seus tipos de dados, do `sampledb` banco de dados do catálogo de dados. `AwsDataCatalog`

```
aws athena get-table-metadata \  
  --catalog-name AwsDataCatalog \  
  --database-name sampledb \  
  --table-name counties
```

Saída:

```
{  
  "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.q1.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.serialization.format": "1",
    "serde.serialization.lib": "com.esri.hadoop.hive.serde.JsonSerde",
    "transient_lastDdlTime": "1593559968"
}
}
}

```

Para obter mais informações, consulte [Mostrando detalhes da tabela: get-table-metadata](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetTableMetadata](#) na Referência de AWS CLI Comandos.

get-work-group

O código de exemplo a seguir mostra como usar get-work-group.

AWS CLI

Para retornar informações sobre um grupo de trabalho

O get-work-group exemplo a seguir retorna informações sobre o AthenaAdmin grupo de trabalho.

```
aws athena get-work-group \
  --work-group AthenaAdmin
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de trabalho](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [GetWorkGroup](#) na Referência de AWS CLI Comandos.

list-data-catalogs

O código de exemplo a seguir mostra como usar `list-data-catalogs`.

AWS CLI

Para listar os catálogos de dados registrados no Athena

O `list-data-catalogs` exemplo a seguir lista os catálogos de dados registrados no Athena.

```
aws athena list-data-catalogs
```

Saída:

```
{
  "DataCatalogsSummary": [
    {
      "CatalogName": "AwsDataCatalog",
      "Type": "GLUE"
    },
    {
      "CatalogName": "cw_logs_catalog",
      "Type": "LAMBDA"
    },
    {
```

```

        "CatalogName": "cw_metrics_catalog",
        "Type": "LAMBDA"
    }
]
}

```

Para obter mais informações, consulte [Listando catálogos registrados: list-data-catalogs](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListDataCatalogs](#) na Referência de AWS CLI Comandos.

list-databases

O código de exemplo a seguir mostra como usar `list-databases`.

AWS CLI

Para listar os bancos de dados em um catálogo de dados

O `list-databases` exemplo a seguir lista os bancos de dados no catálogo `AwsDataCatalog` de dados.

```

aws athena list-databases \
  --catalog-name AwsDataCatalog

```

Saída:

```

{
  "DatabaseList": [
    {
      "Name": "default"
    },
    {
      "Name": "mydatabase"
    },
    {
      "Name": "newdb"
    },
    {
      "Name": "sampledb",
      "Description": "Sample database",
      "Parameters": {
        "CreatedBy": "Athena",

```

```
        "EXTERNAL": "TRUE"
      }
    },
    {
      "Name": "webdata"
    }
  ]
}
```

Para obter mais informações, consulte [Listar bancos de dados em um catálogo: bancos de dados de lista](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListDatabases](#) na Referência de AWS CLI Comandos.

list-named-queries

O código de exemplo a seguir mostra como usar `list-named-queries`.

AWS CLI

Para listar as consultas nomeadas para um grupo de trabalho

O `list-named-queries` exemplo a seguir lista as consultas nomeadas para o grupo AthenaAdmin de trabalho.

```
aws athena list-named-queries \
  --work-group AthenaAdmin
```

Saída:

```
{
  "NamedQueryIds": [
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
  ]
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListNamedQueries](#) na Referência de AWS CLI Comandos.

list-query-executions

O código de exemplo a seguir mostra como usar `list-query-executions`.

AWS CLI

Para listar a consulta IDs das consultas em um grupo de trabalho especificado

O `list-query-executions` exemplo a seguir lista no máximo dez da consulta IDs no AthenaAdmin grupo de trabalho.

```
aws athena list-query-executions \  
  --work-group AthenaAdmin \  
  --max-items 10
```

Saída:

```
{  
  "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="
```

Para obter mais informações, consulte Como [trabalhar com resultados de consultas, arquivos de saída e histórico de consultas](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListQueryExecutions](#) na Referência de AWS CLI Comandos.

list-table-metadata

O código de exemplo a seguir mostra como usar `list-table-metadata`.

AWS CLI

Para listar os metadados das tabelas no banco de dados especificado de um catálogo de dados

O `list-table-metadata` exemplo a seguir retorna informações de metadados para no máximo duas tabelas no `geography` banco de dados do catálogo de `AwsDataCatalog` dados.

```
aws athena list-table-metadata \  
  --catalog-name AwsDataCatalog \  
  --database-name geography \  
  --max-items 2
```

Saída:

```
{  
  "TableMetadataList": [  
    {  
      "Name": "country_codes",  
      "CreateTime": 1586553454.0,  
      "TableType": "EXTERNAL_TABLE",  
      "Columns": [  
        {  
          "Name": "country",  
          "Type": "string",  
          "Comment": "geo id"  
        },  
        {  
          "Name": "alpha-2 code",  
          "Type": "string",  
          "Comment": "geo id2"  
        },  
        {  
          "Name": "alpha-3 code",  
          "Type": "string",  
          "Comment": "state name"  
        },  
        {  
          "Name": "numeric code",  
          "Type": "bigint",  
          "Comment": ""  
        },  
        {  
          "Name": "latitude",
```

```

        "Type": "bigint",
        "Comment": "location (latitude)"
    },
    {
        "Name": "longitude",
        "Type": "bigint",
        "Comment": "location (longitude)"
    }
],
"Parameters": {
    "areColumnsQuoted": "false",
    "classification": "csv",
    "columnsOrdered": "true",
    "delimiter": ",",
    "has_encrypted_data": "false",
    "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
    "location": "s3://awsdoc-example-bucket/csv/countrycode",
    "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.field.delim": ",",
    "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "skip.header.line.count": "1",
    "typeOfData": "file"
}
},
{
    "Name": "county_populations",
    "CreateTime": 1586553446.0,
    "TableType": "EXTERNAL_TABLE",
    "Columns": [
        {
            "Name": "id",
            "Type": "string",
            "Comment": "geo id"
        },
        {
            "Name": "country",

            "Name": "id2",
            "Type": "string",
            "Comment": "geo id2"
        },
    ],
}

```

```

        "Name": "county",
        "Type": "string",
        "Comment": "county name"
    },
    {
        "Name": "state",
        "Type": "string",
        "Comment": "state name"
    },
    {
        "Name": "population estimate 2018",
        "Type": "string",
        "Comment": ""
    }
],
"Parameters": {
    "areColumnsQuoted": "false",
    "classification": "csv",
    "columnsOrdered": "true",
    "delimiter": ",",
    "has_encrypted_data": "false",
    "inputformat": "org.apache.hadoop.mapred.TextInputFormat",
    "location": "s3://awsdoc-example-bucket/csv/CountyPopulation",
    "outputformat":
"org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat",
    "serde.param.field.delim": ",",
    "serde.serialization.lib":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
    "skip.header.line.count": "1",
    "typeOfData": "file"
}
}
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

Para obter mais informações, consulte [Mostrar metadados de todas as tabelas em um banco de dados: list-table-metadata no Guia](#) do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListTableMetadata](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Exemplo 1: Para listar as tags de um grupo de trabalho

O `list-tags-for-resource` exemplo a seguir lista as tags do `Data_Analyst_Group` grupo de trabalho.

```
aws athena list-tags-for-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/  
Data_Analyst_Group
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Division",  
      "Value": "West"  
    },  
    {  
      "Key": "Team",  
      "Value": "Big Data"  
    },  
    {  
      "Key": "Location",  
      "Value": "Seattle"  
    }  
  ]  
}
```

Exemplo 2: Para listar as tags de um catálogo de dados

O `list-tags-for-resource` exemplo a seguir lista as tags do catálogo `dynamo_db_catalog` de dados.

```
aws athena list-tags-for-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog
```


Saída:

```
{
  "Tags": [
    {
      "Key": "Division",
      "Value": "Mountain"
    },
    {
      "Key": "Organization",
      "Value": "Retail"
    },
    {
      "Key": "Product_Line",
      "Value": "Shoes"
    },
    {
      "Key": "Location",
      "Value": "Denver"
    }
  ]
}
```

Para obter mais informações, consulte [Listar as tags de um recurso: list-tags-for-resource](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-work-groups

O código de exemplo a seguir mostra como usar `list-work-groups`.

AWS CLI

Para listar grupos de trabalho

O `list-work-groups` exemplo a seguir lista os grupos de trabalho na conta atual.

```
aws athena list-work-groups
```

Saída:

```
{
```

```

    "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
      }
    ]
  }

```

Para obter mais informações, consulte [Gerenciamento de grupos de trabalho](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [ListWorkGroups](#) na Referência de AWS CLI Comandos.

start-query-execution

O código de exemplo a seguir mostra como usar start-query-execution.

AWS CLI

Exemplo 1: Para executar uma consulta em um grupo de trabalho na tabela especificada no banco de dados e no catálogo de dados especificados

O start-query-execution exemplo a seguir usa o AthenaAdmin grupo de trabalho para executar uma consulta na cloudfront_logs tabela cflogsdatabase no catálogo de AwsDataCatalog dados.

```

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
```

Saída:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

Exemplo 2: Para executar uma consulta que usa um grupo de trabalho especificado para criar um banco de dados no catálogo de dados especificado

O `start-query-execution` exemplo a seguir usa o `AthenaAdmin` grupo de trabalho para criar o banco de dados `newdb` no catálogo `AwsDataCatalog` de dados padrão.

```
aws athena start-query-execution \  
  --query-string "create database if not exists newdb" \  
  --work-group "AthenaAdmin"
```

Saída:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"  
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

Exemplo 3: Para executar uma consulta que cria uma exibição em uma tabela no banco de dados e no catálogo de dados especificados

O `start-query-execution` exemplo a seguir usa uma `SELECT` declaração na `cloudfront_logs` tabela do `cflogsdatabase` para criar a exibição `cf10`.

```
aws athena start-query-execution \  
  --query-string "CREATE OR REPLACE VIEW cf10 AS SELECT * FROM cloudfront_logs  
  Limit 10" \  
  --work-group "AthenaAdmin"
```

```
--query-execution-context Database=cflogsdatabase
```

Saída:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"  
}
```

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [StartQueryExecution](#) na Referência de AWS CLI Comandos.

stop-query-execution

O código de exemplo a seguir mostra como usar `stop-query-execution`.

AWS CLI

Para interromper uma consulta em execução

O `stop-query-execution` exemplo a seguir interrompe a consulta que tem o ID de consulta especificado.

```
aws athena stop-query-execution \  
--query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Este comando não produz saída.

Para obter mais informações, consulte [Execução de SQL consultas usando o Amazon Athena](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [StopQueryExecution](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a um recurso

O `tag-resource` exemplo a seguir adiciona três tags ao catálogo `dynamo_db_catalog` de dados.

```
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
```

Este comando não produz saída. Para ver o resultado, use `aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog`.

Para obter mais informações, consulte [Adicionar tags a um recurso: tag-resource](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove as Focus chaves `Specialization` e e seus valores associados do recurso do catálogo de `dynamo_db_catalog` dados.

```
aws athena untag-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog \  
  --tag-keys Specialization Focus
```

Este comando não produz saída. Para ver os resultados, use o `list-tags-for-resource` comando.

Para obter mais informações, consulte [Removendo tags de um recurso: untag-resource](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-data-catalog

O código de exemplo a seguir mostra como usar `update-data-catalog`.

AWS CLI

Para atualizar um catálogo de dados

O `update-data-catalog` exemplo a seguir atualiza a função Lambda e a descrição do catálogo de `cw_logs_catalog` dados.

```
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
```

Este comando não produz saída. Para ver o resultado, use `aws athena get-data-catalog --name cw_logs_catalog`.

Para obter mais informações, consulte [Atualização de um catálogo: update-data-catalog](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [UpdateDataCatalog](#) na Referência de AWS CLI Comandos.

update-work-group

O código de exemplo a seguir mostra como usar `update-work-group`.

AWS CLI

Para atualizar um grupo de trabalho

O `update-work-group` exemplo a seguir desativa o grupo de `Data_Analyst_Group` trabalho. Os usuários não podem executar ou criar consultas no grupo de trabalho desativado, mas ainda podem visualizar métricas, controles de limite de uso de dados, configurações do grupo de trabalho, histórico de consultas e consultas salvas.

```
aws athena update-work-group \  
  --work-group Data_Analyst_Group \  
  --state DISABLED
```

Este comando não produz saída. Para verificar a alteração no estado, use `aws athena get-work-group --work-group Data_Analyst_Group` e verifique a State propriedade na saída.

Para obter mais informações, consulte [Gerenciamento de grupos de trabalho](#) no Guia do usuário do Amazon Athena.

- Para API obter detalhes, consulte [UpdateWorkGroup](#) na Referência de AWS CLI Comandos.

Exemplos de Auto Scaling usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com Auto Scaling.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

attach-instances

O código de exemplo a seguir mostra como usar `attach-instances`.

AWS CLI

Para anexar uma instância a um grupo do Auto Scaling

Este exemplo anexa a instância especificada ao grupo de Auto Scaling especificado.

```
aws autoscaling attach-instances \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AttachInstances](#) na Referência de AWS CLI Comandos.

attach-load-balancer-target-groups

O código de exemplo a seguir mostra como usar `attach-load-balancer-target-groups`.

AWS CLI

Como anexar um grupo de destino a um grupo do Auto Scaling

Este exemplo anexa o grupo de destino especificado ao grupo do Auto Scaling especificado.

```
aws autoscaling attach-load-balancer-target-groups \  
  --auto-scaling-group-name my-asg \  
  --target-group-arns arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Este comando não produz saída.

Para obter mais informações, consulte [Elastic Load Balancing e Amazon Auto EC2 Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [AttachLoadBalancerTargetGroups](#) na Referência de AWS CLI Comandos.

attach-load-balancers

O código de exemplo a seguir mostra como usar `attach-load-balancers`.

AWS CLI

Para anexar um Classic Load Balancer a um grupo de Auto Scaling

Este exemplo anexa o Classic Load Balancer especificado ao grupo de Auto Scaling especificado.

```
aws autoscaling attach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```


Este comando não produz saída.

Para obter mais informações, consulte [Elastic Load Balancing e Amazon Auto EC2 Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [AttachLoadBalancers](#) na Referência de AWS CLI Comandos.

cancel-instance-refresh

O código de exemplo a seguir mostra como usar `cancel-instance-refresh`.

AWS CLI

Para cancelar uma atualização de instância

O `cancel-instance-refresh` exemplo a seguir cancela uma atualização de instância em andamento para o grupo de Auto Scaling especificado.

```
aws autoscaling cancel-instance-refresh \  
  --auto-scaling-group-name my-asg
```

Saída:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Para obter mais informações, consulte [Cancelar uma atualização de instância no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [CancelInstanceRefresh](#) na Referência de AWS CLI Comandos.

complete-lifecycle-action

O código de exemplo a seguir mostra como usar `complete-lifecycle-action`.

AWS CLI

Para concluir a ação do ciclo de vida

Este exemplo notifica o Amazon EC2 Auto Scaling de que a ação do ciclo de vida especificada foi concluída para que ele possa concluir a inicialização ou o encerramento da instância.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [CompleteLifecycleAction](#) na Referência de AWS CLI Comandos.

create-auto-scaling-group

O código de exemplo a seguir mostra como usar `create-auto-scaling-group`.

AWS CLI

Exemplo 1: como criar um grupo do Auto Scaling

O exemplo de `create-auto-scaling-group` a seguir cria um grupo do Auto Scaling em sub-redes de várias zonas de disponibilidade de uma região. As instâncias são executadas com a versão padrão do modelo de execução especificado. Observe que os padrões são usados na maioria das outras configurações, como nas políticas de encerramento e na configuração de verificação de integridade.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de Auto Scaling no Guia](#) do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: como anexar o Application Load Balancer, o Network Load Balancer ou o Gateway Load Balancer

Este exemplo especifica o ARN de um grupo-alvo para um balanceador de carga que suporta o tráfego esperado. O tipo de verificação de integridade especifica o ELB. Desta forma, quando o Elastic Load Balancing reportar uma instância como não íntegra, o grupo do Auto Scaling a substitui. O comando também define um período de carência de 600 segundos para a verificação de integridade. O período de carência ajuda a evitar o encerramento prematuro de instâncias recém-iniciadas.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Elastic Load Balancing e Amazon Auto EC2 Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 3: como especificar um grupo de posicionamento e usar a versão mais recente do modelo de execução

Este exemplo executa instâncias em um grupo de posicionamento dentro de uma única zona de disponibilidade. Isso pode ser útil para grupos de baixa latência com HPC cargas de trabalho. Esse exemplo também especifica o tamanho mínimo e máximo e a capacidade desejada do grupo.

```
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
```

```
--placement-group my-placement-group \  
--vpc-zone-identifier "subnet-6194ea3b"
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de posicionamento](#) no Guia EC2 do usuário da Amazon para instâncias Linux.

Exemplo 4: como especificar um grupo do Auto Scaling de instância única e usar uma versão específica para iniciar o modelo

Este exemplo cria um grupo do Auto Scaling com capacidade mínima e máxima definida como 1 para impor que uma apenas instância seja executada. O comando também especifica a v1 de um modelo de lançamento no qual o ID de um existente ENI é especificado. Ao usar um modelo de execução que especifica um existente ENI para eth0, você deve especificar uma zona de disponibilidade para o grupo Auto Scaling que corresponda à interface de rede, sem especificar também uma ID de sub-rede na solicitação.

```
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 \  
  --availability-zones us-west-2a
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de Auto Scaling no Guia](#) do usuário do Amazon Auto EC2 Scaling.

Exemplo 5: como especificar uma política de encerramento diferente

Este exemplo cria um grupo do Auto Scaling usando uma configuração de execução e define a política de encerramento para encerrar as instâncias mais antigas primeiro. O comando também aplica uma tag ao grupo e suas instâncias, com uma chave Role e valor de WebServer.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-configuration-name my-lc \  
  --min-size 1 \  
  --tag-key Role \  
  --tag-value WebServer
```

```
--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"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com as políticas de rescisão do Amazon EC2 Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling](#).

Exemplo 6: como especificar um gancho do ciclo de vida de lançamento

Este exemplo a seguir cria um grupo do Auto Scaling com um gancho do ciclo de vida que oferece suporte a uma ação personalizada na inicialização da instância.

```
aws autoscaling create-auto-scaling-group \  
--cli-input-json file://~/config.json
```

Conteúdo do arquivo 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"
  ]}
}
```

Este comando não produz saída.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 7: como especificar um gancho do ciclo de vida de encerramento

Este exemplo a seguir cria um grupo do Auto Scaling com um gancho do ciclo de vida que oferece suporte a uma ação personalizada no encerramento da instância.

```
aws autoscaling create-auto-scaling-group \
  --cli-input-json file://~/config.json
```

Conteúdo de 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"
}
```

Este comando não produz saída.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 8: como especificar uma política de encerramento personalizada

Este exemplo cria um grupo de Auto Scaling que especifica uma política personalizada de encerramento da função Lambda que informa ao Amazon Auto Scaling EC2 quais instâncias podem ser encerradas em escala segura.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma política de rescisão personalizada com o Lambda no Guia](#) do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [CreateAutoScalingGroup](#) na Referência de AWS CLI Comandos.

create-launch-configuration

O código de exemplo a seguir mostra como usar create-launch-configuration.

AWS CLI

Exemplo 1: Para criar uma configuração de inicialização

Este exemplo cria uma configuração de inicialização simples.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma configuração de lançamento](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Exemplo 2: Para criar uma configuração de inicialização com um grupo de segurança, um key pair e um script de bootstrapping

Este exemplo cria uma configuração de inicialização com um grupo de segurança, um key pair e um script de bootstrapping contido nos dados do usuário.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma configuração de lançamento](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Exemplo 3: Para criar uma configuração de lançamento com uma IAM função

Este exemplo cria uma configuração de execução com o nome do perfil da instância de uma IAM função.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --iam-instance-profile my-autoscaling-role
```

Este comando não produz saída.

Para obter mais informações, consulte a [IAM função dos aplicativos que são executados em EC2 instâncias da Amazon](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Exemplo 4: Para criar uma configuração de lançamento com o monitoramento detalhado ativado

Este exemplo cria uma configuração de lançamento com monitoramento EC2 detalhado ativado, que envia EC2 métricas CloudWatch em períodos de 1 minuto.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --instance-monitoring Enabled=true
```

Este comando não produz saída.

Para obter mais informações, consulte [Configuração do monitoramento para instâncias do Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 5: Para criar uma configuração de execução que inicie instâncias spot

Este exemplo cria uma configuração de lançamento que usa Instâncias Spot como a única opção de compra.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --spot-price "0.50"
```

Este comando não produz saída.

Para obter mais informações, consulte [Solicitação de instâncias spot no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 6: Para criar uma configuração de execução usando uma EC2 instância

Este exemplo cria uma configuração de execução com base nos atributos de uma instância existente. Ele substitui a localização do posicionamento e se um endereço IP público é definido incluindo as opções e. `--placement-tenancy` `--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 \  
  --no-associate-public-ip-address
```

```
--placement-tenancy dedicated
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma configuração de execução usando uma EC2 instância](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Exemplo 7: Para criar uma configuração de lançamento com um mapeamento de dispositivos de blocos para um EBS volume da Amazon

Este exemplo cria uma configuração de lançamento com um mapeamento de dispositivos de blocos para um EBS gp3 volume da Amazon com o nome do dispositivo /dev/sdh e um tamanho de volume de 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"}}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [EBS](#) a Amazon EC2 Auto Scaling API Reference.

Para obter informações sobre a sintaxe para citar valores JSON de parâmetros formatados, consulte [Usando aspas com cadeias de caracteres no Guia do usuário da](#) interface de linha de comando AWS CLI comando.AWS

Exemplo 8: Para criar uma configuração de execução com um mapeamento de dispositivos de blocos para um volume de armazenamento de instâncias

Este exemplo cria uma configuração de execução com ephemeral1 um volume de armazenamento de instância com o nome do dispositivo/dev/sdc.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --block-device-mappings '[{"DeviceName":"/dev/sdc", "VirtualName":"ephemeral1"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [BlockDeviceMapping](#) Amazon EC2 Auto Scaling API Reference.

Para obter informações sobre a sintaxe para citar valores JSON de parâmetros formatados, consulte [Usando aspas com cadeias de caracteres no Guia do usuário da](#) interface de linha de AWS CLI comando.AWS

Exemplo 9: Para criar uma configuração de inicialização e impedir que um dispositivo de bloco se conecte no momento da inicialização

Este exemplo cria uma configuração de inicialização que suprime um dispositivo de bloco especificado pelo mapeamento do dispositivo de bloco do AMI (por exemplo,/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":""}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [BlockDeviceMapping](#) Amazon EC2 Auto Scaling API Reference.

Para obter informações sobre a sintaxe para citar valores JSON de parâmetros formatados, consulte [Usando aspas com cadeias de caracteres no Guia do usuário da](#) interface de linha de AWS CLI comando.AWS

- Para API obter detalhes, consulte [CreateLaunchConfiguration](#) na Referência de AWS CLI Comandos.

create-or-update-tags

O código de exemplo a seguir mostra como usar `create-or-update-tags`.

AWS CLI

Para criar ou atualizar tags para um grupo de Auto Scaling

Este exemplo adiciona duas tags ao grupo de Auto Scaling especificado.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar grupos e instâncias do Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [CreateOrUpdateTags](#) na Referência de AWS CLI Comandos.

delete-auto-scaling-group

O código de exemplo a seguir mostra como usar delete-auto-scaling-group.

AWS CLI

Exemplo 1: como excluir o grupo do Auto Scaling especificado

Este exemplo exclui o grupo do Auto Scaling especificado.

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir sua infraestrutura de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: como forçar a exclusão do grupo do Auto Scaling especificado

Use a opção `--force-delete` para excluir o grupo do Auto Scaling sem precisar esperar que as instâncias do grupo sejam encerradas.

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir sua infraestrutura de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DeleteAutoScalingGroup](#) na Referência de AWS CLI Comandos.

delete-launch-configuration

O código de exemplo a seguir mostra como usar `delete-launch-configuration`.

AWS CLI

Para excluir uma configuração de inicialização

Este exemplo exclui a configuração de inicialização especificada.

```
aws autoscaling delete-launch-configuration \  
  --launch-configuration-name my-launch-config
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir sua infraestrutura de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DeleteLaunchConfiguration](#) na Referência de AWS CLI Comandos.

delete-lifecycle-hook

O código de exemplo a seguir mostra como usar `delete-lifecycle-hook`.

AWS CLI

Para excluir um gancho de ciclo de vida

Este exemplo exclui o gancho de ciclo de vida especificado.

```
aws autoscaling delete-lifecycle-hook \  
  --lifecycle-hook-name my-lifecycle-hook \  
  --auto-scaling-group-name my-asg
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteLifecycleHook](#) na Referência de AWS CLI Comandos.

delete-notification-configuration

O código de exemplo a seguir mostra como usar delete-notification-configuration.

AWS CLI

Para excluir uma notificação do Auto Scaling

Este exemplo exclui a notificação especificada do grupo de Auto Scaling especificado.

```
aws autoscaling delete-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir a configuração de notificação](#) no Guia do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [DeleteNotificationConfiguration](#) na Referência de AWS CLI Comandos.

delete-policy

O código de exemplo a seguir mostra como usar delete-policy.

AWS CLI

Para excluir uma política de escalabilidade

Este exemplo exclui a política de escalabilidade especificada.

```
aws autoscaling delete-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name alb1000-target-tracking-scaling-policy
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

delete-scheduled-action

O código de exemplo a seguir mostra como usar `delete-scheduled-action`.

AWS CLI

Para excluir uma ação agendada de um grupo do Auto Scaling

Este exemplo exclui a ação agendada especificada do grupo de Auto Scaling especificado.

```
aws autoscaling delete-scheduled-action \  
  --auto-scaling-group-name my-asg \  
  --scheduled-action-name my-scheduled-action
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteScheduledAction](#) na Referência de AWS CLI Comandos.

delete-tags

O código de exemplo a seguir mostra como usar `delete-tags`.

AWS CLI

Para excluir uma tag de um grupo de Auto Scaling

Este exemplo exclui a tag especificada do grupo de Auto Scaling especificado.

```
aws autoscaling delete-tags \  
  --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar grupos e instâncias do Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DeleteTags](#) na Referência de AWS CLI Comandos.

delete-warm-pool

O código de exemplo a seguir mostra como usar `delete-warm-pool`.

AWS CLI

Exemplo 1: Para excluir uma piscina aquecida

O exemplo a seguir exclui a piscina aquecida do grupo de Auto Scaling especificado.

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg
```

Este comando não produz saída.

Para obter mais informações, consulte [Warm Pools for Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: Para forçar a exclusão de uma piscina aquecida

Para excluir o pool aquecido sem esperar que suas instâncias terminem, use a `--force-delete` opção.

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

Este comando não produz saída.

Para obter mais informações, consulte [Warm Pools for Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DeleteWarmPool](#) na Referência de AWS CLI Comandos.

describe-account-limits

O código de exemplo a seguir mostra como usar `describe-account-limits`.

AWS CLI

Para descrever os limites da sua conta Amazon EC2 Auto Scaling

Este exemplo descreve os limites do Amazon EC2 Auto Scaling para sua AWS conta.

```
aws autoscaling describe-account-limits
```


Saída:

```
{
  "NumberOfLaunchConfigurations": 5,
  "MaxNumberOfLaunchConfigurations": 100,
  "NumberOfAutoScalingGroups": 3,
  "MaxNumberOfAutoScalingGroups": 20
}
```

Para obter mais informações, consulte as [cotas do serviço Amazon EC2 Auto Scaling no Guia do usuário](#) do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [DescribeAccountLimits](#) na Referência de AWS CLI Comandos.

describe-adjustment-types

O código de exemplo a seguir mostra como usar `describe-adjustment-types`.

AWS CLI

Para descrever os tipos de ajuste de escala disponíveis

Este exemplo descreve os tipos de ajuste disponíveis.

```
aws autoscaling describe-adjustment-types
```

Saída:

```
{
  "AdjustmentTypes": [
    {
      "AdjustmentType": "ChangeInCapacity"
    },
    {
      "AdjustmentType": "ExactCapacity"
    },
    {
      "AdjustmentType": "PercentChangeInCapacity"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Tipos de ajuste de escalabilidade no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [DescribeAdjustmentTypes](#) na Referência de AWS CLI Comandos.

describe-auto-scaling-groups

O código de exemplo a seguir mostra como usar `describe-auto-scaling-groups`.

AWS CLI

Exemplo 1: como descrever o grupo do Auto Scaling especificado

Este exemplo descreve o grupo do Auto Scaling especificado.

```
aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-name my-asg
```

Saída:

```
{
  "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": []
    }
]
}

```

Exemplo 2: como descrever os primeiros 100 grupos do Auto Scaling especificados

Este exemplo descreve os grupos do Auto Scaling especificados. Ele permite especificar até cem nomes de grupos.

```

aws autoscaling describe-auto-scaling-groups \
  --max-items 100 \

```

```
--auto-scaling-group-name "group1" "group2" "group3" "group4"
```

Consulte um exemplo de saída no exemplo 1.

Exemplo 3: como descrever um grupo do Auto Scaling na região especificada

Este exemplo descreve até 75 grupos do Auto Scaling na região especificada.

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 75 \  
  --region us-east-1
```

Consulte um exemplo de saída no exemplo 1.

Exemplo 4: como descrever o número especificado do grupo do Auto Scaling

Use a opção `--max-items` para retornar um número específico de grupos do Auto Scaling.

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 1
```

Consulte um exemplo de saída no exemplo 1.

Se a saída incluir um campo `NextToken`, há mais grupos. Para obter os grupos adicionais, use o valor desse campo com a opção `--starting-token` em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-auto-scaling-groups \  
  --starting-token Z3M3LMPEXAMPLE
```

Consulte um exemplo de saída no exemplo 1.

Exemplo 5: Para descrever grupos de Auto Scaling que usam configurações de inicialização

Este exemplo usa a `--query` opção para descrever grupos de Auto Scaling que usam configurações de inicialização.

```
aws autoscaling describe-auto-scaling-groups \  
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Saída:

```
[
  {
    "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": []
  }
]
```

```
}  
]
```

Para obter mais informações, consulte [AWS CLISaída do filtro](#) no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DescribeAutoScalingGroups](#) na Referência de AWS CLI Comandos.

describe-auto-scaling-instances

O código de exemplo a seguir mostra como usar `describe-auto-scaling-instances`.

AWS CLI

Exemplo 1: como descrever uma ou mais instâncias

Este exemplo descreve a instância especificada.

```
aws autoscaling describe-auto-scaling-instances \  
  --instance-ids i-06905f55584de02da
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Exemplo 2: como descrever uma ou mais instâncias

Este exemplo usa a opção `--max-items` para especificar quantas instâncias devem ser retornadas pela chamada.

```
aws autoscaling describe-auto-scaling-instances \  
  --max-items 1
```

Se a saída incluir um campo `NextToken`, há mais instâncias. Para obter as instâncias adicionais, use o valor desse campo com a opção `--starting-token` em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-auto-scaling-instances \  
  --starting-token Z3M3LMPEXAMPLE
```

Consulte um exemplo de saída no exemplo 1.

Exemplo 3: Para descrever instâncias que usam configurações de execução

Este exemplo usa a opção `--query` para descrever instâncias que usam configurações de execução.

```
aws autoscaling describe-auto-scaling-instances \  
  --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

Saída:

```
[  
  {  
    "InstanceId": "i-088c57934a6449037",  
    "InstanceType": "t2.micro",  
    "AutoScalingGroupName": "my-asg",  
    "AvailabilityZone": "us-west-2c",  
    "LifecycleState": "InService",  
    "HealthStatus": "HEALTHY",  
    "LaunchConfigurationName": "my-lc",  
    "ProtectedFromScaleIn": false  
  }  
]
```

Para obter mais informações, consulte [AWS CLISaída do filtro](#) no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DescribeAutoScalingInstances](#) na Referência de AWS CLI Comandos.

describe-auto-scaling-notification-types

O código de exemplo a seguir mostra como usar `describe-auto-scaling-notification-types`.

AWS CLI

Para descrever os tipos de notificação disponíveis

Este exemplo descreve os tipos de notificação disponíveis.

```
aws autoscaling describe-auto-scaling-notification-types
```

Saída:

```
{
  "AutoScalingNotificationTypes": [
    "autoscaling:EC2_INSTANCE_LAUNCH",
    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
    "autoscaling:EC2_INSTANCE_TERMINATE",
    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
    "autoscaling:TEST_NOTIFICATION"
  ]
}
```

Para obter mais informações, consulte [Receber SNS notificações da Amazon quando seu grupo de Auto Scaling aumenta](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DescribeAutoScalingNotificationTypes](#) na Referência de AWS CLI Comandos.

describe-instance-refreshes

O código de exemplo a seguir mostra como usar `describe-instance-refreshes`.

AWS CLI

Para descrever as atualizações de instâncias

O `describe-instance-refreshes` exemplo a seguir retorna uma descrição de todas as solicitações de atualização de instância para o grupo de Auto Scaling especificado, incluindo a mensagem de status e (se disponível) o motivo do status.

```
aws autoscaling describe-instance-refreshes \  
  --auto-scaling-group-name my-asg
```

Saída:

```
{  
  "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"
    }
}
]
}

```

Para obter mais informações, consulte [Verifique o status de uma atualização de instância](#) no Guia do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [DescribeInstanceRefreshes](#) na Referência de AWS CLI Comandos.

describe-launch-configurations

O código de exemplo a seguir mostra como usar `describe-launch-configurations`.

AWS CLI

Exemplo 1: Para descrever a configuração de lançamento especificada

Este exemplo descreve a configuração de lançamento especificada.

```
aws autoscaling describe-launch-configurations \
  --launch-configuration-names my-launch-config
```

Saída:

```
{
  "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"
    }
  }
]
}

```

Exemplo 2: Para descrever um número específico de configurações de lançamento

Para retornar um número específico de configurações de lançamento, use a `--max-items` opção.

```

aws autoscaling describe-launch-configurations \
  --max-items 1

```

Se a saída incluir um NextToken campo, haverá mais configurações de inicialização. Para obter as configurações adicionais de inicialização, use o valor desse campo com a `--starting-token` opção em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-launch-configurations \
  --starting-token Z3M3LMPEXAMPLE
```

- Para API obter detalhes, consulte [DescribeLaunchConfigurations](#) na Referência de AWS CLI Comandos.

describe-lifecycle-hook-types

O código de exemplo a seguir mostra como usar `describe-lifecycle-hook-types`.

AWS CLI

Para descrever os tipos de ganchos de ciclo de vida disponíveis

Este exemplo descreve os tipos de ganchos de ciclo de vida disponíveis.

```
aws autoscaling describe-lifecycle-hook-types
```

Saída:

```
{
  "LifecycleHookTypes": [
    "autoscaling:EC2_INSTANCE_LAUNCHING",
    "autoscaling:EC2_INSTANCE_TERMINATING"
  ]
}
```

- Para API obter detalhes, consulte [DescribeLifecycleHookTypes](#) na Referência de AWS CLI Comandos.

describe-lifecycle-hooks

O código de exemplo a seguir mostra como usar `describe-lifecycle-hooks`.

AWS CLI

Para descrever seus ganchos de ciclo de vida

Este exemplo descreve os ganchos do ciclo de vida do grupo de Auto Scaling especificado.

```
aws autoscaling describe-lifecycle-hooks \  
  --auto-scaling-group-name my-asg
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeLifecycleHooks](#) na Referência de AWS CLI Comandos.

describe-load-balancer-target-groups

O código de exemplo a seguir mostra como usar `describe-load-balancer-target-groups`.

AWS CLI

Para descrever os grupos-alvo do balanceador de carga para um grupo do Auto Scaling

Este exemplo descreve os grupos-alvo do balanceador de carga anexados ao grupo de Auto Scaling especificado.

```
aws autoscaling describe-load-balancer-target-groups \
  --auto-scaling-group-name my-asg
```

Saída:

```
{
  "LoadBalancerTargetGroups": [
    {
      "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
      "State": "Added"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLoadBalancerTargetGroups](#) na Referência de AWS CLI Comandos.

describe-load-balancers

O código de exemplo a seguir mostra como usar `describe-load-balancers`.

AWS CLI

Para descrever os balanceadores de carga clássicos para um grupo de Auto Scaling

Este exemplo descreve os Classic Load Balancers para o grupo de Auto Scaling especificado.

```
aws autoscaling describe-load-balancers \
  --auto-scaling-group-name my-asg
```

Saída:

```
{
  "LoadBalancers": [
    {
      "State": "Added",
      "LoadBalancerName": "my-load-balancer"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLoadBalancers](#) na Referência de AWS CLI Comandos.

describe-metric-collection-types

O código de exemplo a seguir mostra como usar `describe-metric-collection-types`.

AWS CLI

Para descrever os tipos de coleta de métricas disponíveis

Este exemplo descreve os tipos de coleta de métricas disponíveis.

```
aws autoscaling describe-metric-collection-types
```

Saída:

```
{
  "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"
  }
]
}
```

Para obter mais informações, consulte as [métricas de grupo do Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling](#).

- Para API obter detalhes, consulte [DescribeMetricCollectionTypes](#) na Referência de AWS CLI Comandos.

describe-notification-configurations

O código de exemplo a seguir mostra como usar `describe-notification-configurations`.

AWS CLI

Exemplo 1: Para descrever as configurações de notificação de um grupo especificado

Este exemplo descreve as configurações de notificação para o grupo de Auto Scaling especificado.

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-asg
```


Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Receber SNS notificações da Amazon quando seu grupo de Auto Scaling aumenta](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 1: Para descrever um número específico de configurações de notificação

Para retornar um número específico de configurações de notificação, use o `max-items` parâmetro.

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-auto-scaling-group \
  --max-items 1
```

Saída:

```
{
  "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 a saída incluir um NextToken campo, haverá mais configurações de notificação. Para obter as configurações adicionais de notificação, use o valor desse campo com o `starting-token` parâmetro em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-asg \
  --starting-token Z3M3LMPEXAMPLE
```

Para obter mais informações, consulte [Receber SNS notificações da Amazon quando seu grupo de Auto Scaling aumenta](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DescribeNotificationConfigurations](#) na Referência de AWS CLI Comandos.

describe-policies

O código de exemplo a seguir mostra como usar `describe-policies`.

AWS CLI

Exemplo 1: Para descrever as políticas de escalabilidade de um grupo especificado

Este exemplo descreve as políticas de escalabilidade para o grupo de Auto Scaling especificado.

```
aws autoscaling describe-policies \
  --auto-scaling-group-name my-asg
```

Saída:

```
{
  "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
}
]
}

```

Para obter mais informações, consulte [Escalabilidade dinâmica no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 2: Para descrever as políticas de escalabilidade de um nome especificado

Para retornar políticas de escalabilidade específicas, use a `--policy-names` opção.

```

aws autoscaling describe-policies \
  --auto-scaling-group-name my-asg \
  --policy-names cpu40-target-tracking-scaling-policy

```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Escalabilidade dinâmica no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 3: Para descrever várias políticas de escalabilidade

Para retornar um número específico de políticas, use a `--max-items` opção.

```

aws autoscaling describe-policies \
  --auto-scaling-group-name my-asg \

```

```
--max-items 1
```

Consulte um exemplo de saída no exemplo 1.

Se a saída incluir um NextToken campo, use o valor desse campo com a `--starting-token` opção em uma chamada subsequente para obter as políticas adicionais.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-token Z3M3LMPEXAMPLE
```

Para obter mais informações, consulte [Escalabilidade dinâmica no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [DescribePolicies](#) na Referência de AWS CLI Comandos.

describe-scaling-activities

O código de exemplo a seguir mostra como usar `describe-scaling-activities`.

AWS CLI

Exemplo 1: como descrever as ações de escalabilidade do grupo especificado

Este exemplo descreve as ações de escalabilidade do grupo do Auto Scaling especificado.

```
aws autoscaling describe-scaling-activities \  
--auto-scaling-group-name my-asg
```

Saída:

```
{  
  "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\\\"}"
    }
]
}

```

Para obter mais informações, consulte [Verificar uma atividade de escalabilidade para um grupo de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: como descrever as ações de escalabilidade de um grupo excluído

Adicione a opção `--include-deleted-groups` para descrever as ações de escalabilidade depois que o grupo do Auto Scaling for excluído.

```

aws autoscaling describe-scaling-activities \
  --auto-scaling-group-name my-asg \
  --include-deleted-groups

```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Solucionar problemas do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 3: como descrever um número especificado de ações de escalabilidade

Use a opção `--max-items` para retornar um número específico de ações.

```

aws autoscaling describe-scaling-activities \
  --max-items 1

```

Saída:

```

{
  "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 a saída incluir um campo `NextToken`, há mais ações. Para obter as ações adicionais, use o valor desse campo com a opção `--starting-token` em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-scaling-activities \  
  --starting-token Z3M3LMPEXAMPLE
```

Para obter mais informações, consulte [Verificar uma atividade de escalabilidade para um grupo de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DescribeScalingActivities](#) na Referência de AWS CLI Comandos.

describe-scaling-process-types

O código de exemplo a seguir mostra como usar `describe-scaling-process-types`.

AWS CLI

Para descrever os tipos de processo disponíveis

Este exemplo descreve os tipos de processo disponíveis.

```
aws autoscaling describe-scaling-process-types
```

Saída:

```
{  
  "Processes": [  
    {  
      "ProcessName": "AZRebalance"  
    },  
    {  
      "ProcessName": "AddToLoadBalancer"  
    },  
    {  
      "ProcessName": "AlarmNotification"  
    },  
    {  
      "ProcessName": "HealthCheck"  
    },  
    {
```



```
        "ProcessName": "InstanceRefresh"
      },
      {
        "ProcessName": "Launch"
      },
      {
        "ProcessName": "ReplaceUnhealthy"
      },
      {
        "ProcessName": "ScheduledActions"
      },
      {
        "ProcessName": "Terminate"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Suspensão e retomada dos processos de escalabilidade no Guia do usuário](#) do Amazon EC2Auto Scaling.

- Para API obter detalhes, consulte [DescribeScalingProcessTypes](#) na Referência de AWS CLI Comandos.

describe-scheduled-actions

O código de exemplo a seguir mostra como usar `describe-scheduled-actions`.

AWS CLI

Exemplo 1: Para descrever todas as ações agendadas

Este exemplo descreve todas as suas ações agendadas.

```
aws autoscaling describe-scheduled-actions
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 2: Para descrever ações agendadas para o grupo especificado

Para descrever as ações agendadas para um grupo específico do Auto Scaling, use a `--auto-scaling-group-name` opção.

```

aws autoscaling describe-scheduled-actions \
  --auto-scaling-group-name my-asg

```

Saída:

```

{
  "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"
    }
  ]
}

```

```
}
```

Para obter mais informações, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 3: Para descrever a ação agendada especificada

Para descrever uma ação agendada específica, use a `--scheduled-action-names` opção.

```
aws autoscaling describe-scheduled-actions \  
  --scheduled-action-names my-recurring-action
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 4: Para descrever ações agendadas com um horário de início especificado

Para descrever as ações agendadas que começam em um horário específico, use a `--start-time` opção.

```
aws autoscaling describe-scheduled-actions \  
  --start-time
```

```
--start-time "2023-12-01T04:00:00Z"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 5: Para descrever ações agendadas que terminam em um horário especificado

Para descrever as ações agendadas que terminam em um horário específico, use a `--end-time` opção.

```
aws autoscaling describe-scheduled-actions \
  --end-time "2023-12-01T04:00:00Z"
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 6: Para descrever um número específico de ações programadas

Para retornar um número específico de ações agendadas, use a `--max-items` opção.

```

aws autoscaling describe-scheduled-actions \
  --auto-scaling-group-name my-asg \
  --max-items 1

```

Saída:

```

{
  "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 a saída incluir um NextToken campo, haverá mais ações agendadas. Para obter as ações agendadas adicionais, use o valor desse campo com a `--starting-token` opção em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

Para obter mais informações, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [DescribeScheduledActions](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever todas as tags

Este exemplo descreve todas as suas tags.

```
aws autoscaling describe-tags
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte Como [marcar grupos e instâncias do Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: Para descrever tags para um grupo especificado

Para descrever as tags de um grupo específico do Auto Scaling, use a `--filters` opção.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

Para obter mais informações, consulte Como [marcar grupos e instâncias do Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 3: Para descrever o número especificado de tags

Para retornar um número específico de tags, use a `--max-items` opção.

```
aws autoscaling describe-tags \
  --max-items 1
```

Se a saída incluir um `NextToken` campo, haverá mais tags. Para obter as tags adicionais, use o valor desse campo com a `--starting-token` opção em uma chamada subsequente da seguinte maneira.

```
aws autoscaling describe-tags \
  --filters Name=auto-scaling-group,Values=my-asg \
  --starting-token Z3M3LMPEXAMPLE
```

Para obter mais informações, consulte Como [marcar grupos e instâncias do Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

describe-termination-policy-types

O código de exemplo a seguir mostra como usar `describe-termination-policy-types`.

AWS CLI

Para descrever os tipos de política de rescisão disponíveis

Este exemplo descreve os tipos de política de rescisão disponíveis.

```
aws autoscaling describe-termination-policy-types
```

Saída:

```
{
  "TerminationPolicyTypes": [
    "AllocationStrategy",
    "ClosestToNextInstanceHour",
    "Default",
    "NewestInstance",
    "OldestInstance",
    "OldestLaunchConfiguration",
    "OldestLaunchTemplate"
  ]
}
```

Para obter mais informações, consulte Como [controlar quais instâncias do Auto Scaling terminam durante a escalabilidade no Guia do usuário](#) do Amazon EC2Auto Scaling.

- Para API obter detalhes, consulte [DescribeTerminationPolicyTypes](#) na Referência de AWS CLI Comandos.

describe-warm-pool

O código de exemplo a seguir mostra como usar `describe-warm-pool`.

AWS CLI

Para descrever uma piscina aquecida

Este exemplo descreve a piscina aquecida para o grupo de Auto Scaling especificado.

```
aws autoscaling describe-warm-pool \
  --auto-scaling-group-name my-asg
```


Saída:

```
{
  "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": "Warmed:Pending",
      "HealthStatus": "Healthy",
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-00a731f6e9fa48610",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "Version": "6"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Warm Pools for Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DescribeWarmPool](#) na Referência de AWS CLI Comandos.

detach-instances

O código de exemplo a seguir mostra como usar detach-instances.

AWS CLI

Para separar uma instância de um grupo do Auto Scaling

Este exemplo separa a instância especificada do grupo de Auto Scaling especificado.

```
aws autoscaling detach-instances \  
  --instance-ids i-030017cfa84b20135 \  
  --auto-scaling-group-name my-asg \  
  --should-decrement-desired-capacity
```

Saída:

```
{  
  "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\"}"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DetachInstances](#) na Referência de AWS CLI Comandos.

detach-load-balancer-target-groups

O código de exemplo a seguir mostra como usar detach-load-balancer-target-groups.

AWS CLI

Para separar um grupo-alvo do balanceador de carga de um grupo do Auto Scaling

Este exemplo separa o grupo-alvo do balanceador de carga especificado do grupo de Auto Scaling especificado.

```
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
```

Esse comando não produz nenhuma saída

Para obter mais informações, consulte Como [anexar um balanceador de carga ao seu grupo de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DetachLoadBalancerTargetGroups](#) na Referência de AWS CLI Comandos.

detach-load-balancers

O código de exemplo a seguir mostra como usar detach-load-balancers.

AWS CLI

Para separar um Classic Load Balancer de um grupo de Auto Scaling

Este exemplo separa o Classic Load Balancer especificado do grupo de Auto Scaling especificado.

```
aws autoscaling detach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```

Este comando não produz saída.

Para obter mais informações, consulte Como [anexar um balanceador de carga ao seu grupo de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DetachLoadBalancers](#) na Referência de AWS CLI Comandos.

disable-metrics-collection

O código de exemplo a seguir mostra como usar disable-metrics-collection.

AWS CLI

Como desabilitar a coleta de métricas de um grupo do Auto Scaling

Este exemplo desativa a coleta da métrica `GroupDesiredCapacity` do grupo do Auto Scaling especificado.

```
aws autoscaling disable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity
```

Este comando não produz saída.

Para obter mais informações, consulte [CloudWatch Métricas de monitoramento para seus grupos e instâncias do Auto Scaling no Guia](#) do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [DisableMetricsCollection](#) na Referência de AWS CLI Comandos.

enable-metrics-collection

O código de exemplo a seguir mostra como usar `enable-metrics-collection`.

AWS CLI

Exemplo 1: como habilitar a coleta de métricas do grupo do Auto Scaling

Este exemplo habilita a coleta de dados do grupo do Auto Scaling especificado.

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --granularity "1Minute"
```

Este comando não produz saída.

Para obter mais informações, consulte [CloudWatch Métricas de monitoramento para seus grupos e instâncias do Auto Scaling no Guia](#) do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: como coletar dados da métrica especificada de um grupo de Auto Scaling

Use a opção `--metrics` para coletar dados de uma métrica específica.

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity --granularity "1Minute"
```

Este comando não produz saída.

Para obter mais informações, consulte [CloudWatch Métricas de monitoramento para seus grupos e instâncias do Auto Scaling no Guia](#) do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [EnableMetricsCollection](#) na Referência de AWS CLI Comandos.

enter-standby

O código de exemplo a seguir mostra como usar `enter-standby`.

AWS CLI

Para mover instâncias para o modo de espera

Este exemplo coloca a instância especificada no modo de espera. Isso é útil para atualizar ou solucionar problemas de uma instância que está atualmente em serviço.

```
aws autoscaling enter-standby \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg \  
  --should-decrement-desired-capacity
```

Saída:

```
{  
  "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\"}"  
    }  
  ]  
}
```

Para obter mais informações, consulte o [ciclo de vida da instância do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto Scaling. EC2

- Para API obter detalhes, consulte [EnterStandby](#) na Referência de AWS CLI Comandos.

execute-policy

O código de exemplo a seguir mostra como usar `execute-policy`.

AWS CLI

Para executar uma política de escalabilidade

Este exemplo executa a política de escalabilidade nomeada `my-step-scale-out-policy` para o grupo de Auto Scaling especificado.

```
aws autoscaling execute-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name my-step-scale-out-policy \  
  --metric-value 95 \  
  --breach-threshold 80
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas de escalabilidade simples e por etapas no Guia do usuário do Amazon EC2 Auto Scaling](#).

- Para API obter detalhes, consulte [ExecutePolicy](#) na Referência de AWS CLI Comandos.

exit-standby

O código de exemplo a seguir mostra como usar `exit-standby`.

AWS CLI

Para mover instâncias para fora do modo de espera

Este exemplo move a instância especificada para fora do modo de espera.

```
aws autoscaling exit-standby \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg
```

Saída:

```
{
  "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\\\"}"
    }
  ]
}
```

Para obter mais informações, consulte [Remoção temporária de instâncias do seu grupo de Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling](#).

- Para API obter detalhes, consulte [ExitStandby](#) na Referência de AWS CLI Comandos.

put-lifecycle-hook

O código de exemplo a seguir mostra como usar `put-lifecycle-hook`.

AWS CLI

Exemplo 1: Para criar um gancho de ciclo de vida

Este exemplo cria um gancho de ciclo de vida que será invocado em qualquer instância recém-lançada, com um tempo limite de 4800 segundos. Isso é útil para manter as instâncias em um estado de espera até que os scripts de dados do usuário terminem ou para invocar uma função AWS Lambda usando 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
```

Este comando não produz saída. Se já existir um gancho de ciclo de vida com o mesmo nome, ele será substituído pelo novo gancho de ciclo de vida.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: Enviar uma mensagem de SNS e-mail da Amazon para notificá-lo sobre as transições de estado da instância

Este exemplo cria um gancho de ciclo de vida com o SNS tópico e a IAM função da Amazon para usar para receber notificações na inicialização da instância.

```
aws autoscaling put-lifecycle-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-hook-name my-launch-hook \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING \  
  --notification-target-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \  
  --role-arn arn:aws:iam::123456789012:role/my-auto-scaling-role
```

Este comando não produz saída.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 3: Para publicar uma mensagem em uma SQS fila da Amazon

Este exemplo cria um gancho de ciclo de vida que publica uma mensagem com metadados na fila especificada da Amazon. 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"
```

Este comando não produz saída.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [PutLifecycleHook](#) na Referência de AWS CLI Comandos.

put-notification-configuration

O código de exemplo a seguir mostra como usar `put-notification-configuration`.

AWS CLI

Para adicionar uma notificação

Este exemplo adiciona a notificação especificada ao grupo de Auto Scaling especificado.

```
aws autoscaling put-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic \  
  --notification-type autoscaling:TEST_NOTIFICATION
```

Este comando não produz saída.

Para obter mais informações, consulte [Receber SNS notificações da Amazon quando seu grupo de Auto Scaling aumenta](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [PutNotificationConfiguration](#) na Referência de AWS CLI Comandos.

put-scaling-policy

O código de exemplo a seguir mostra como usar `put-scaling-policy`.

AWS CLI

Para adicionar uma política de escalabilidade de rastreamento de metas a um grupo de Auto Scaling

O `put-scaling-policy` exemplo a seguir aplica uma política de escalabilidade de rastreamento de metas ao grupo de Auto Scaling especificado. A saída contém os nomes ARNs e dos dois CloudWatch alarmes criados em seu nome. Se já existir uma política de escalabilidade com o mesmo nome, ela será substituída pela nova política de escalabilidade.

```
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
```

Conteúdo de config.json:

```
{  
  "TargetValue": 1000.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ALBRequestCountPerTarget",  
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-  
group/943f017f100becff"  
  }  
}
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais exemplos, consulte [Exemplos de políticas de escalabilidade para a interface de linha de AWS comando \(AWS CLI\) no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [PutScalingPolicy](#) na Referência de AWS CLI Comandos.

put-scheduled-update-group-action

O código de exemplo a seguir mostra como usar `put-scheduled-update-group-action`.

AWS CLI

Exemplo 1: Para adicionar uma ação agendada a um grupo de Auto Scaling

Este exemplo adiciona a ação agendada especificada ao grupo de Auto Scaling especificado.

```
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
```

Este comando não produz saída. Se uma ação agendada com o mesmo nome já existir, ela será substituída pela nova ação agendada.

Para obter mais exemplos, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 2: Para especificar uma agenda recorrente

Este exemplo cria uma ação programada para escalar em uma programação recorrente que está programada para ser executada às 00:30 horas nos primeiros dias de janeiro, junho e dezembro de cada ano.

```
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
```

Este comando não produz saída. Se uma ação agendada com o mesmo nome já existir, ela será substituída pela nova ação agendada.

Para obter mais exemplos, consulte [Escalabilidade programada no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [PutScheduledUpdateGroupAction](#) na Referência de AWS CLI Comandos.

put-warm-pool

O código de exemplo a seguir mostra como usar `put-warm-pool`.

AWS CLI

Para criar uma piscina aquecida

O exemplo a seguir cria uma piscina aquecida para o grupo de Auto Scaling especificado.

```
aws autoscaling put-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --min-size 2
```

Este comando não produz saída. Se já existir uma piscina aquecida, ela será atualizada.

Para obter mais informações, consulte [Warm Pools for Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [PutWarmPool](#) na Referência de AWS CLI Comandos.

record-lifecycle-action-heartbeat

O código de exemplo a seguir mostra como usar `record-lifecycle-action-heartbeat`.

AWS CLI

Para registrar a pulsação de uma ação do ciclo de vida

Este exemplo registra uma pulsação de ação do ciclo de vida para manter a instância em um estado pendente.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte os [ganchos do ciclo de vida do Amazon EC2 Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [RecordLifecycleActionHeartbeat](#) na Referência de AWS CLI Comandos.

resume-processes

O código de exemplo a seguir mostra como usar `resume-processes`.

AWS CLI

Para retomar processos suspensos

Este exemplo retoma o processo de escalonamento suspenso especificado para o grupo de Auto Scaling especificado.

```
aws autoscaling resume-processes \  
  --auto-scaling-group-name my-asg \  
  --scaling-processes AlarmNotification
```

Este comando não produz saída.

Para obter mais informações, consulte [Suspensão e retomada dos processos de escalabilidade no Guia do usuário](#) do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [ResumeProcesses](#) na Referência de AWS CLI Comandos.

rollback-instance-refresh

O código de exemplo a seguir mostra como usar `rollback-instance-refresh`.

AWS CLI

Para reverter uma atualização de instância

O `rollback-instance-refresh` exemplo a seguir reverte uma atualização de instância em andamento para o grupo de Auto Scaling especificado.

```
aws autoscaling rollback-instance-refresh \  
  --auto-scaling-group-name my-asg \  
  --instance-refresh-id my-refresh-id
```

```
--auto-scaling-group-name my-asg
```

Saída:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Para obter mais informações, consulte [Desfazer alterações com uma reversão no Guia do usuário do Amazon EC2 Auto Scaling](#).

- Para API obter detalhes, consulte [RollbackInstanceRefresh](#) na Referência de AWS CLI Comandos.

set-desired-capacity

O código de exemplo a seguir mostra como usar `set-desired-capacity`.

AWS CLI

Como definir a capacidade desejada para um grupo do Auto Scaling

Este exemplo define a capacidade desejada para o grupo do Auto Scaling especificado.

```
aws autoscaling set-desired-capacity \  
  --auto-scaling-group-name my-asg \  
  --desired-capacity 2 \  
  --honor-cooldown
```

Esse comando retornará à solicitação, se houver êxito.

- Para API obter detalhes, consulte [SetDesiredCapacity](#) na Referência de AWS CLI Comandos.

set-instance-health

O código de exemplo a seguir mostra como usar `set-instance-health`.

AWS CLI

Para definir o status de integridade de uma instância

Este exemplo define o status de integridade da instância especificada como `Unhealthy`.

```
aws autoscaling set-instance-health \  
  --instance-id i-061c63c5eb45f0416 \  
  --health-status Unhealthy
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetInstanceHealth](#) na Referência de AWS CLI Comandos.

set-instance-protection

O código de exemplo a seguir mostra como usar `set-instance-protection`.

AWS CLI

Exemplo 1: Para ativar a configuração de proteção de instância para uma instância

Este exemplo habilita a proteção de instância para a instância especificada.

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg --protected-from-scale-in
```

Este comando não produz saída.

Exemplo 2: Para desativar a configuração de proteção de instância para uma instância

Este exemplo desativa a proteção de instância para a instância especificada.

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg \  
  --no-protected-from-scale-in
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetInstanceProtection](#) na Referência de AWS CLI Comandos.

start-instance-refresh

O código de exemplo a seguir mostra como usar `start-instance-refresh`.

AWS CLI

Exemplo 1: para iniciar uma atualização de instância usando parâmetros da linha de comando

O `start-instance-refresh` exemplo a seguir inicia uma atualização da instância usando argumentos de linha de comando. O `preferences` parâmetro opcional especifica um `InstanceWarmup` de 60 segundos e um `MinHealthyPercentage` de 50 porcentagem.

```
aws autoscaling start-instance-refresh \  
  --auto-scaling-group-name my-asg \  
  --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```

Saída:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Para obter mais informações, consulte [Iniciar uma atualização de instância no Guia](#) do usuário do Amazon EC2 Auto Scaling.

Exemplo 2: para iniciar uma atualização de instância usando um arquivo JSON

O `start-instance-refresh` exemplo a seguir inicia uma atualização de instância usando um JSON arquivo. Você pode especificar o grupo Auto Scaling e definir a configuração e as preferências desejadas em um JSON arquivo, conforme mostrado no exemplo a seguir.

```
aws autoscaling start-instance-refresh \  
  --cli-input-json file://config.json
```

Conteúdo de `config.json`:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "DesiredConfiguration": {  
    "LaunchTemplate": {  
      "LaunchTemplateId": "lt-068f72b729example",  
      "Version": "$Default"  
    }  
  },  
}
```



```
"Preferences": {
  "InstanceWarmup": 60,
  "MinHealthyPercentage": 50,
  "AutoRollback": true,
  "ScaleInProtectedInstances": Ignore,
  "StandbyInstances": Terminate
}
```

Saída:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Para obter mais informações, consulte [Iniciar uma atualização de instância no Guia](#) do usuário do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [StartInstanceRefresh](#) na Referência de AWS CLI Comandos.

suspend-processes

O código de exemplo a seguir mostra como usar `suspend-processes`.

AWS CLI

Para suspender os processos do Auto Scaling

Este exemplo suspende o processo de escalonamento especificado para o grupo de Auto Scaling especificado.

```
aws autoscaling suspend-processes \
  --auto-scaling-group-name my-asg \
  --scaling-processes AlarmNotification
```

Este comando não produz saída.

Para obter mais informações, consulte [Suspensão e retomada dos processos de escalabilidade no Guia do usuário](#) do Amazon EC2 Auto Scaling.

- Para API obter detalhes, consulte [SuspendProcesses](#) na Referência de AWS CLI Comandos.

terminate-instance-in-auto-scaling-group

O código de exemplo a seguir mostra como usar `terminate-instance-in-auto-scaling-group`.

AWS CLI

Como encerrar uma instância em um grupo do Auto Scaling

Este exemplo encerra a instância especificada do grupo do Auto Scaling especificado sem atualizar o tamanho do grupo. O Amazon EC2 Auto Scaling inicia uma instância substituta após o término da instância especificada.

```
aws autoscaling terminate-instance-in-auto-scaling-group \
  --instance-id i-061c63c5eb45f0416 \
  --no-should-decrement-desired-capacity
```

Saída:

```
{
  "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\"}"
    }
  ]
}
```

- Para API obter detalhes, consulte [TerminateInstanceInAutoScalingGroup](#) na Referência de AWS CLI Comandos.

update-auto-scaling-group

O código de exemplo a seguir mostra como usar `update-auto-scaling-group`.

AWS CLI

Exemplo 1: como atualizar os limites de tamanho de um grupo do Auto Scaling

Este exemplo atualiza o grupo do Auto Scaling especificado com um tamanho mínimo de 2 e máximo de 10.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --min-size 2 \  
  --max-size 10
```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo limites de capacidade para seu grupo de Auto Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 2: como adicionar verificações de integridade do Elastic Load Balancing e especificar quais zonas de disponibilidade e sub-redes usar

Este exemplo atualiza o grupo do Auto Scaling especificado para adicionar verificações de integridade do Elastic Load Balancing. Esse comando também atualiza o valor de `--vpc-zone-identifier` com uma lista de sub-redes IDs em várias zonas de disponibilidade.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Elastic Load Balancing e Amazon Auto EC2 Scaling](#) no Guia do usuário do Amazon Auto EC2 Scaling.

Exemplo 3: como atualizar o grupo de posicionamento e a política de encerramento

Este exemplo atualiza o grupo de posicionamento e a política de encerramento que devem ser usados.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --availability-zones "us-east-1a,us-east-1b" \  
  --termination-protection true
```

```
--placement-group my-placement-group \  
--termination-policies "OldestInstance"
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de Auto Scaling no Guia](#) do usuário do Amazon Auto EC2 Scaling.

Exemplo 4: como usar a versão mais recente do modelo de execução

Este exemplo atualiza o grupo do Auto Scaling especificado para que use a versão mais recente do modelo de execução especificado.

```
aws autoscaling update-auto-scaling-group \  
--auto-scaling-group-name my-asg \  
--launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

Este comando não produz saída.

Para obter mais informações, consulte [Modelos de lançamento](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Exemplo 5: como usar uma versão específica do modelo de execução

Este exemplo atualiza o grupo do Auto Scaling especificado para que use uma versão específica do modelo de execução em vez da versão mais recente ou padrão.

```
aws autoscaling update-auto-scaling-group \  
--auto-scaling-group-name my-asg \  
--launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Este comando não produz saída.

Para obter mais informações, consulte [Modelos de lançamento](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Exemplo 6: como definir uma política de instâncias mistas e habilitar o rebalanceamento de capacidade

Este exemplo atualiza o grupo do Auto Scaling especificado para que use uma política de instâncias mistas e permita o rebalanceamento de capacidade. Essa estrutura permite especificar

grupos com capacidades spot e sob demanda e usar modelos de execução diferentes para arquiteturas diferentes.

```
aws autoscaling update-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Conteúdo de 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"  
    }  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de Auto Scaling com vários tipos de instância e opções de compra](#) no Guia do usuário do Amazon Auto EC2 Scaling.

- Para API obter detalhes, consulte [UpdateAutoScalingGroup](#) na Referência de AWS CLI Comandos.

Exemplos de planos de Auto Scaling usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com planos de Auto Scaling.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-scaling-plan

O código de exemplo a seguir mostra como usar `create-scaling-plan`.

AWS CLI

Para criar um plano de escalabilidade

O `create-scaling-plan` exemplo a seguir cria um plano de escalabilidade chamado `my-scaling-plan` usando um JSON arquivo já criado (chamado `config.json`). A estrutura do plano de escalonamento inclui uma instrução de escalonamento para um grupo de Auto Scaling chamado `my-asg`. Ele especifica a propriedade `TagFilters` como a origem do aplicativo e permite a escalabilidade preditiva e a escalabilidade dinâmica.

```
aws autoscaling-plans create-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --cli-input-json file://~/config.json
```

Conteúdo do arquivo 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
        }
      ]
    }
  ]
}
```

Saída:

```
{
```

```
"ScalingPlanVersion": 1
}
```

Para obter mais informações, consulte o Guia do [usuário do AWS Auto Scaling](#).

- Para API obter detalhes, consulte [CreateScalingPlan](#) na Referência de AWS CLI Comandos.

delete-scaling-plan

O código de exemplo a seguir mostra como usar delete-scaling-plan.

AWS CLI

Para excluir um plano de escalabilidade

O delete-scaling-plan exemplo a seguir exclui o plano de escalabilidade especificado.

```
aws autoscaling-plans delete-scaling-plan \
  --scaling-plan-name my-scaling-plan \
  --scaling-plan-version 1
```

Este comando não produz saída.

Para obter mais informações, consulte o Guia do [usuário do AWS Auto Scaling](#).

- Para API obter detalhes, consulte [DeleteScalingPlan](#) na Referência de AWS CLI Comandos.

describe-scaling-plan-resources

O código de exemplo a seguir mostra como usar describe-scaling-plan-resources.

AWS CLI

Para descrever os recursos escaláveis para um plano de escalabilidade

O describe-scaling-plan-resources exemplo a seguir exibe detalhes sobre o único recurso escalável (um grupo de Auto Scaling) associado ao plano de escalabilidade especificado.

```
aws autoscaling-plans describe-scaling-plan-resources \
  --scaling-plan-name my-scaling-plan \
  --scaling-plan-version 1
```


Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [O que é o AWS Auto Scaling?](#) no Guia do usuário do AWS Auto Scaling.

- Para API obter detalhes, consulte [DescribeScalingPlanResources](#) na Referência de AWS CLI Comandos.

describe-scaling-plans

O código de exemplo a seguir mostra como usar `describe-scaling-plans`.

AWS CLI

Para descrever um plano de escalabilidade

O `describe-scaling-plans` exemplo a seguir exibe os detalhes do plano de escalabilidade especificado.

```
aws autoscaling-plans describe-scaling-plans \  
--scaling-plan-names scaling-plan-with-asg-and-ddb
```

Saída:

```
{  
  "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"
}
]
}

```

Para obter mais informações, consulte [O que é o AWS Auto Scaling?](#) no Guia do usuário do AWS Auto Scaling.

- Para API obter detalhes, consulte [DescribeScalingPlans](#) na Referência de AWS CLI Comandos.

get-scaling-plan-resource-forecast-data

O código de exemplo a seguir mostra como usar `get-scaling-plan-resource-forecast-data`.

AWS CLI

Para recuperar dados de previsão de carga

Este exemplo recupera dados de previsão de carga para um recurso escalável (um grupo de Auto Scaling) associado ao plano de escalabilidade especificado.

```

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"

```

Saída:

```
{
  "Datapoints": [...]
}
```

Para obter mais informações, consulte [O que é o AWS Auto Scaling](#) no Guia do usuário do Auto AWS Scaling.

- Para API obter detalhes, consulte [GetScalingPlanResourceForecastData](#) na Referência de AWS CLI Comandos.

update-scaling-plan

O código de exemplo a seguir mostra como usar `update-scaling-plan`.

AWS CLI

Para atualizar um plano de escalabilidade

O `update-scaling-plan` exemplo a seguir modifica a métrica de escalabilidade de um grupo de Auto Scaling no plano de escalabilidade especificado.

```
aws autoscaling-plans update-scaling-plan \
  --scaling-plan-name my-scaling-plan \
  --scaling-plan-version 1 \
  --scaling-instructions
  '{"ScalableDimension":"autoscaling:autoScalingGroup:DesiredCapacity","ResourceId":"autoScal
my-asg","ServiceNamespace":"autoscaling","TargetTrackingConfigurations":
[{"PredefinedScalingMetricSpecification":
  {"PredefinedScalingMetricType":"ALBRequestCountPerTarget","ResourceLabel":"app/my-
alb/f37c06a68c1748aa/targetgroup/my-target-
group/6d4ea56ca2d6a18d"},"TargetValue":40.0}],"MinCapacity": 1,"MaxCapacity": 10}'
```

Este comando não produz saída.

Para obter mais informações, consulte [O que é o AWS Auto Scaling?](#) no Guia do usuário do AWS Auto Scaling.

- Para API obter detalhes, consulte [UpdateScalingPlan](#) na Referência de AWS CLI Comandos.

AWS Backup exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Backup.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-backup-plan

O código de exemplo a seguir mostra como usar `create-backup-plan`.

AWS CLI

Para criar um plano de backup

O `create-backup-plan` exemplo a seguir cria o plano de backup especificado com retenção de 35 dias.

```
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}}]}"
```

Saída:

```
{
  "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": "ZjQ2ZTI5YWQtZDg5Yi00MzYzLWJmZTAzMDE1Mzh1MDhjYjEz"  
}
```

Para obter mais informações, consulte [Criação de um plano de backup](#) no Guia do desenvolvedor de AWS backup.

- Para API obter detalhes, consulte [CreateBackupPlan](#) na Referência de AWS CLI Comandos.

create-backup-vault

O código de exemplo a seguir mostra como usar `create-backup-vault`.

AWS CLI

Para criar um cofre de backup

O `create-backup-vault` exemplo a seguir cria um cofre de backup com o nome especificado.

```
aws backup create-backup-vault  
--backup-vault-name sample-vault
```

Este comando não produz saída. Saída:

```
{  
  "BackupVaultName": "sample-vault",  
  "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-  
vault",  
  "CreationDate": 1568928338.385  
}
```

Para obter mais informações, consulte [Criação de um Cofre de Backup](#) no Guia do Desenvolvedor AWS de Backup.

- Para API obter detalhes, consulte [CreateBackupVault](#) na Referência de AWS CLI Comandos.

get-backup-plan-from-template

O código de exemplo a seguir mostra como usar `get-backup-plan-from-template`.

AWS CLI

Para obter um plano de backup existente a partir de um modelo

O `get-backup-plan-from-template` exemplo a seguir obtém um plano de backup existente a partir de um modelo que especifica um backup diário com retenção de 35 dias.

```
aws backup get-backup-plan-from-template \  
  --backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

Saída:

```
{  
  "BackupPlanDocument": {  
    "Rules": [  
      {  
        "RuleName": "DailyBackups",  
        "ScheduleExpression": "cron(0 5 ? * * *)",  
        "StartWindowMinutes": 480,  
        "Lifecycle": {  
          "DeleteAfterDays": 35  
        }  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Criação de um plano de backup](#) no Guia do desenvolvedor de AWS backup.

- Para API obter detalhes, consulte [GetBackupPlanFromTemplate](#) na Referência de AWS CLI Comandos.

get-backup-plan

O código de exemplo a seguir mostra como usar `get-backup-plan`.

AWS CLI

Para obter os detalhes de um plano de backup

O `get-backup-plan` exemplo a seguir exibe os detalhes do plano de backup especificado.

```
aws backup get-backup-plan \  
  --backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"
```


Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Criação de um plano de backup](#) no Guia do desenvolvedor de AWS backup.

- Para API obter detalhes, consulte [GetBackupPlan](#) na Referência de AWS CLI Comandos.

list-backup-jobs

O código de exemplo a seguir mostra como usar `list-backup-jobs`.

AWS CLI

Exemplo 1: Para listar todas as tarefas de backup

O `list-backup-jobs` exemplo a seguir retorna metadados sobre suas tarefas de backup em sua AWS conta.

aws backup list-backup-jobs

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Criando um backup](#) no Guia do desenvolvedor de AWS backup.

Exemplo 2: Para listar as tarefas de backup concluídas

O `list-backup-jobs` exemplo a seguir retorna metadados sobre suas tarefas de backup concluídas em sua AWS conta.

```
aws backup list-backup-jobs \  
  --by-state COMPLETED
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Criando um backup](#) no Guia do desenvolvedor de AWS backup.

- Para API obter detalhes, consulte [ListBackupJobs](#) na Referência de AWS CLI Comandos.

AWS Batch exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Batch.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-job

O código de exemplo a seguir mostra como usar `cancel-job`.

AWS CLI

Para cancelar um trabalho

Este exemplo cancela um trabalho com o ID de trabalho especificado.

Comando:

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --  
reason "Canceling job."
```

- Para API obter detalhes, consulte [CancelJob](#) na Referência de AWS CLI Comandos.

create-compute-environment

O código de exemplo a seguir mostra como usar `create-compute-environment`.

AWS CLI

Para criar um ambiente computacional gerenciado com instâncias sob demanda

Este exemplo cria um ambiente computacional gerenciado com tipos específicos de instância C4 que são executados sob demanda. O ambiente computacional é chamado de OnDemand C4.

Comando:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/C4OnDemand.json
```

JSONformato de arquivo:

```
{
  "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"
}
```

Saída:

```
{
  "computeEnvironmentName": "C4OnDemand",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

Para criar um ambiente computacional gerenciado com instâncias spot

Este exemplo cria um ambiente de computação gerenciado com o tipo de instância M4 que é lançado quando o preço da oferta spot é igual ou inferior a 20% do preço sob demanda do tipo de instância. O ambiente computacional é chamado de M4Spot.

Comando:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
M4Spot.json
```

JSONformato de arquivo:

```
{
  "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"
}
```

Saída:

```
{
  "computeEnvironmentName": "M4Spot",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/M4Spot"
}
```

- Para API obter detalhes, consulte [CreateComputeEnvironment](#) na Referência de AWS CLI Comandos.

create-job-queue

O código de exemplo a seguir mostra como usar create-job-queue.

AWS CLI

Para criar uma fila de trabalhos de baixa prioridade com um único ambiente computacional

Este exemplo cria uma fila de trabalhos chamada LowPriority que usa o ambiente de computação M4Spot.

Comando:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
LowPriority.json
```

JSONformato de arquivo:

```
{
  "jobQueueName": "LowPriority",
```

```
"state": "ENABLED",
"priority": 10,
"computeEnvironmentOrder": [
  {
    "order": 1,
    "computeEnvironment": "M4Spot"
  }
]
```

Saída:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",
  "jobQueueName": "LowPriority"
}
```

Para criar uma fila de trabalhos de alta prioridade com dois ambientes computacionais

Este exemplo cria uma fila de trabalhos chamada HighPriority que usa o ambiente de OnDemand computação C4 com uma ordem de 1 e o ambiente de computação M4Spot com uma ordem de 2. O programador tentará primeiro colocar trabalhos no ambiente OnDemand computacional C4.

Comando:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/HighPriority.json
```

JSONformato de arquivo:

```
{
  "jobQueueName": "HighPriority",
  "state": "ENABLED",
  "priority": 1,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "C4OnDemand"
    },
    {
      "order": 2,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```



```
}  
]  
}
```

Saída:

```
{  
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",  
  "jobQueueName": "HighPriority"  
}
```

- Para API obter detalhes, consulte [CreateJobQueue](#) na Referência de AWS CLI Comandos.

delete-compute-environment

O código de exemplo a seguir mostra como usar `delete-compute-environment`.

AWS CLI

Para excluir um ambiente computacional

Este exemplo exclui o ambiente OnDemand computacional P2.

Comando:

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- Para API obter detalhes, consulte [DeleteComputeEnvironment](#) na Referência de AWS CLI Comandos.

delete-job-queue

O código de exemplo a seguir mostra como usar `delete-job-queue`.

AWS CLI

Para excluir uma fila de trabalhos

Este exemplo exclui a fila de GPGPU trabalhos.

Comando:

```
aws batch delete-job-queue --job-queue GPGPU
```

- Para API obter detalhes, consulte [DeleteJobQueue](#) na Referência de AWS CLI Comandos.

deregister-job-definition

O código de exemplo a seguir mostra como usar `deregister-job-definition`.

AWS CLI

Para cancelar o registro de uma definição de cargo

Este exemplo cancela o registro de uma definição de trabalho chamada `sleep10`.

Comando:

```
aws batch deregister-job-definition --job-definition sleep10
```

- Para API obter detalhes, consulte [DeregisterJobDefinition](#) na Referência de AWS CLI Comandos.

describe-compute-environments

O código de exemplo a seguir mostra como usar `describe-compute-environments`.

AWS CLI

Como descrever um ambiente de computação

Este exemplo descreve o ambiente OnDemand computacional P2.

Comando:

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

Saída:

```
{  
  "computeEnvironments": [  
    {  
      "computeEnvironmentName": "P2OnDemand",  
      "computeEnvironmentType": "ON_DEMAND",  
      "state": "AVAILABLE",  
      "provisioningConfiguration": {  
        "image": "amzn2-ami-aws-ec2-20190820",  
        "instanceProfile": "AWSBatchEC2Profile",  
        "instanceType": "m5.xlarge",  
        "keyPair": "AWSBatchKeyPair",  
        "securityGroups": [ "sg-4a7c1111" ],  
        "subnets": [ "subnet-4a7c1111" ],  
        "vpcSubnetFilter": {  
          "vpc": "vpc-4a7c1111",  
          "subnets": [ "subnet-4a7c1111" ]  
        }  
      }  
    }  
  ]  
}
```

```

    {
      "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeComputeEnvironments](#) na Referência de AWS CLI Comandos.

describe-job-definitions

O código de exemplo a seguir mostra como usar `describe-job-definitions`.

AWS CLI

Para descrever as definições de tarefas ativas

Este exemplo descreve todas as suas definições de trabalho ativas.

Comando:

```
aws batch describe-job-definitions --status ACTIVE
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeJobDefinitions](#) na Referência de AWS CLI Comandos.

describe-job-queues

O código de exemplo a seguir mostra como usar `describe-job-queues`.

AWS CLI

Para descrever uma fila de trabalhos

Este exemplo descreve a fila `HighPriority` de trabalhos.

Comando:

```
aws batch describe-job-queues --job-queues HighPriority
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeJobQueues](#) na Referência de AWS CLI Comandos.

describe-jobs

O código de exemplo a seguir mostra como usar `describe-jobs`.

AWS CLI

Para descrever um trabalho

O `describe-jobs` exemplo a seguir descreve um trabalho com o ID de trabalho especificado.

```
aws batch describe-jobs \  
  --jobs bcf0b186-a532-4122-842e-2ccab8d54efb
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeJobs](#) na Referência de AWS CLI Comandos.

list-jobs

O código de exemplo a seguir mostra como usar `list-jobs`.

AWS CLI

Para listar trabalhos em execução

Este exemplo lista os trabalhos em execução na fila de HighPriority trabalhos.

Comando:

```
aws batch list-jobs --job-queue HighPriority
```

Saída:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
    }
  ]
}
```

Para listar trabalhos enviados

Este exemplo lista os trabalhos na fila de HighPriority trabalhos que estão no status do SUBMITTED trabalho.

Comando:

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

Saída:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
    }
  ]
}
```

```
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListJobs](#) na Referência de AWS CLI Comandos.

register-job-definition

O código de exemplo a seguir mostra como usar `register-job-definition`.

AWS CLI

Para registrar uma definição de cargo

Este exemplo registra uma definição de tarefa para uma tarefa de contêiner simples.

Comando:

```
aws batch register-job-definition --job-definition-name sleep30 --type container --  
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":  
[ "sleep", "30"]}'
```

Saída:

```
{  
  "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/  
sleep30:1",  
  "jobDefinitionName": "sleep30",  
  "revision": 1  
}
```

- Para API obter detalhes, consulte [RegisterJobDefinition](#) na Referência de AWS CLI Comandos.

submit-job

O código de exemplo a seguir mostra como usar `submit-job`.

AWS CLI

Para enviar um trabalho

Este exemplo envia um trabalho de contêiner simples chamado `example` para a fila de `HighPriority` trabalhos.

Comando:

```
aws batch submit-job --job-name example --job-queue HighPriority --job-  
definition sleep60
```

Saída:

```
{  
  "jobName": "example",  
  "jobId": "876da822-4198-45f2-a252-6cea32512ea8"  
}
```

- Para API obter detalhes, consulte [SubmitJob](#) na Referência de AWS CLI Comandos.

terminate-job

O código de exemplo a seguir mostra como usar `terminate-job`.

AWS CLI

Para encerrar um trabalho

Este exemplo encerra um trabalho com o ID de trabalho especificado.

Comando:

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --  
reason "Terminating job."
```

- Para API obter detalhes, consulte [TerminateJob](#) na Referência de AWS CLI Comandos.

update-compute-environment

O código de exemplo a seguir mostra como usar `update-compute-environment`.

AWS CLI

Para atualizar um ambiente computacional

Este exemplo desativa o ambiente OnDemand computacional P2 para que ele possa ser excluído.

Comando:

```
aws batch update-compute-environment --compute-environment P2OnDemand --state DISABLED
```

Saída:

```
{
  "computeEnvironmentName": "P2OnDemand",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/P2OnDemand"
}
```

- Para API obter detalhes, consulte [UpdateComputeEnvironment](#) na Referência de AWS CLI Comandos.

update-job-queue

O código de exemplo a seguir mostra como usar `update-job-queue`.

AWS CLI

Para atualizar uma fila de trabalhos

Este exemplo desativa uma fila de trabalhos para que ela possa ser excluída.

Comando:

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

Saída:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",
  "jobQueueName": "GPGPU"
}
```

- Para API obter detalhes, consulte [UpdateJobQueue](#) na Referência de AWS CLI Comandos.

AWS Budgets exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Budgets.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-budget

O código de exemplo a seguir mostra como usar `create-budget`.

AWS CLI

Para criar um orçamento de custo e uso

O `create-budget` comando a seguir cria um orçamento de custo e uso.

```
aws budgets create-budget \  
  --account-id 111122223333 \  
  --budget file://budget.json \  
  --notifications-with-subscribers file://notifications-with-subscribers.json
```

Conteúdo de `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"
}
```

Conteúdo de notifications-with-subscribers.json:

```
[
  {
    "Notification": {
      "ComparisonOperator": "GREATER_THAN",
      "NotificationType": "ACTUAL",
      "Threshold": 80,
      "ThresholdType": "PERCENTAGE"
    },
    "Subscribers": [
      {
        "Address": "example@example.com",
        "SubscriptionType": "EMAIL"
      }
    ]
  }
]
```

- Para API obter detalhes, consulte [CreateBudget](#) na Referência de AWS CLI Comandos.

create-notification

O código de exemplo a seguir mostra como usar create-notification.

AWS CLI

Para criar uma notificação para o orçamento de custo e uso especificado

Este exemplo cria uma notificação para o orçamento de custo e uso especificado.

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
```

- Para API obter detalhes, consulte [CreateNotification](#) na Referência de AWS CLI Comandos.

create-subscriber

O código de exemplo a seguir mostra como usar create-subscriber.

AWS CLI

Para criar um assinante para uma notificação associada a um orçamento de custo e uso

Este exemplo cria um assinante para a notificação especificada.

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
```

- Para API obter detalhes, consulte [CreateSubscriber](#) na Referência de AWS CLI Comandos.

delete-budget

O código de exemplo a seguir mostra como usar delete-budget.

AWS CLI

Para excluir um orçamento de custo e uso

Este exemplo exclui o orçamento de custo e uso especificado.

Comando:

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- Para API obter detalhes, consulte [DeleteBudget](#) na Referência de AWS CLI Comandos.

delete-notification

O código de exemplo a seguir mostra como usar delete-notification.

AWS CLI

Para excluir uma notificação de um orçamento

Este exemplo exclui a notificação especificada do orçamento especificado.

Comando:

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example Budget" --  
notification NotificationType=ACTUAL, ComparisonOperator=GREATER_THAN, Threshold=80, ThresholdD
```

- Para API obter detalhes, consulte [DeleteNotification](#) na Referência de AWS CLI Comandos.

delete-subscriber

O código de exemplo a seguir mostra como usar delete-subscriber.

AWS CLI

Para excluir um assinante de uma notificação

Este exemplo exclui o assinante especificado da notificação especificada.

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
```

- Para API obter detalhes, consulte [DeleteSubscriber](#) na Referência de AWS CLI Comandos.

describe-budget

O código de exemplo a seguir mostra como usar `describe-budget`.

AWS CLI

Para recuperar um orçamento associado a uma conta

Este exemplo recupera o orçamento de custo e uso especificado.

Comando:

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

Saída:

```
{
  "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"
    ]
  }
}
```

- Para API obter detalhes, consulte [DescribeBudget](#) na Referência de AWS CLI Comandos.

describe-budgets

O código de exemplo a seguir mostra como usar `describe-budgets`.

AWS CLI

Para recuperar os orçamentos associados a uma conta

Este exemplo recupera os orçamentos de custo e uso de uma conta.

Comando:


```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

Saída:

```
{
  "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"
  ]
}

```

- Para API obter detalhes, consulte [DescribeBudgets](#) na Referência de AWS CLI Comandos.

describe-notifications-for-budget

O código de exemplo a seguir mostra como usar `describe-notifications-for-budget`.

AWS CLI

Para recuperar as notificações de um orçamento

Este exemplo recupera as notificações de um orçamento de custo e uso.

Comando:

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-name "Example Budget" --max-results 5
```

Saída:

```
{
  "Notifications": [
    {
      "Threshold": 80.0,
      "ComparisonOperator": "GREATER_THAN",
      "NotificationType": "ACTUAL"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeNotificationsForBudget](#) na Referência de AWS CLI Comandos.

describe-subscribers-for-notification

O código de exemplo a seguir mostra como usar `describe-subscribers-for-notification`.

AWS CLI

Para recuperar os assinantes para uma notificação de orçamento

Este exemplo recupera os assinantes de uma notificação de orçamento de custo e uso.

Comando:

```
aws budgets describe-subscribers-for-notification --  
account-id 111122223333 --budget-name "Example Budget" --  
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=COST --  
--max-results 5
```

Saída:

```
{  
  "Subscribers": [  
    {  
      "SubscriptionType": "EMAIL",  
      "Address": "example2@example.com"  
    },  
    {  
      "SubscriptionType": "EMAIL",  
      "Address": "example@example.com"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeSubscribersForNotification](#) na Referência de AWS CLI Comandos.

update-budget

O código de exemplo a seguir mostra como usar update-budget.

AWS CLI

Para substituir um orçamento por um orçamento de custo e uso

Este exemplo substitui um orçamento de custo e uso por um novo orçamento.

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"
}
```

- Para API obter detalhes, consulte [UpdateBudget](#) na Referência de AWS CLI Comandos.

update-notification

O código de exemplo a seguir mostra como usar update-notification.

AWS CLI

Para substituir uma notificação por um orçamento de custo e uso

Este exemplo substitui uma notificação de 80% para um orçamento de custo e uso por uma notificação de 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
```

- Para API obter detalhes, consulte [UpdateNotification](#) na Referência de AWS CLI Comandos.

update-subscriber

O código de exemplo a seguir mostra como usar update-subscriber.

AWS CLI

Para substituir um assinante por um orçamento de custo e uso

Este exemplo substitui o assinante por um orçamento de custo e uso.

Comando:

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENT --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- Para API obter detalhes, consulte [UpdateSubscriber](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Chime usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Chime.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-phone-number-with-user

O código de exemplo a seguir mostra como usar `associate-phone-number-with-user`.

AWS CLI

Para associar um número de telefone a um usuário

O `associate-phone-number-with-user` exemplo a seguir associa o número de telefone especificado a um usuário.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de números de telefone de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [AssociatePhoneNumberWithUser](#) na Referência de AWS CLI Comandos.

associate-phone-numbers-with-voice-connector-group

O código de exemplo a seguir mostra como usar `associate-phone-numbers-with-voice-connector-group`.

AWS CLI

Para associar números de telefone a um grupo do Amazon Chime Voice Connector

O `associate-phone-numbers-with-voice-connector-group` exemplo a seguir associa os números de telefone especificados a um grupo Amazon Chime Voice Connector.

```
aws chime associate-phone-numbers-with-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901 \  
  --e164-phone-numbers " +12065550100" "+12065550101" \  
  --force-associate
```

Saída:

```
{  
  "PhoneNumberErrors": []  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos do Amazon Chime Voice Connector no Guia](#) de Administração do Amazon Chime.

- Para API obter detalhes, consulte [AssociatePhoneNumbersWithVoiceConnectorGroup](#) na Referência de AWS CLI Comandos.

`associate-phone-numbers-with-voice-connector`

O código de exemplo a seguir mostra como usar `associate-phone-numbers-with-voice-connector`.

AWS CLI

Para associar números de telefone a um Amazon Chime Voice Connector

O `associate-phone-numbers-with-voice-connector` exemplo a seguir associa os números de telefone especificados a um Amazon Chime Voice Connector.

```
aws chime associate-phone-numbers-with-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --e164-phone-numbers " +12065550100" "+12065550101" \  
  --force-associate
```

Saída:

```
{
  "PhoneNumberErrors": []
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [AssociatePhoneNumbersWithVoiceConnector](#) na Referência de AWS CLI Comandos.

associate-signin-delegate-groups-with-account

O código de exemplo a seguir mostra como usar `associate-signin-delegate-groups-with-account`.

AWS CLI

Para associar grupos de delegados de login

O `associate-signin-delegate-groups-with-account` exemplo a seguir associa o grupo de delegados de login especificado à conta especificada do Amazon Chime.

```
aws chime associate-signin-delegate-groups-with-account \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --signin-delegate-groups GroupName=my_users
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de acesso e permissões de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [AssociateSigninDelegateGroupsWithAccount](#) na Referência de AWS CLI Comandos.

batch-create-room-membership

O código de exemplo a seguir mostra como usar `batch-create-room-membership`.

AWS CLI

Para criar várias associações de salas

O `batch-create-room-membership` exemplo a seguir adiciona vários usuários a uma sala de bate-papo como membros da sala de bate-papo. Ele também atribui funções de administrador e membro aos usuários.

```
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"
```

Saída:

```
{  
  "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": []  
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [BatchCreateRoomMembership](#) na Referência de AWS CLI Comandos.

batch-delete-phone-number

O código de exemplo a seguir mostra como usar `batch-delete-phone-number`.

AWS CLI

Para excluir vários números de telefone

O `batch-delete-phone-number` exemplo a seguir exclui todos os números de telefone especificados.

```
aws chime batch-delete-phone-number \  
  --phone-number-ids "%2B12065550100" "%2B12065550101"
```

Este comando não produz saída. Saída:

```
{  
  "PhoneNumberErrors": []  
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [BatchDeletePhoneNumber](#) na Referência de AWS CLI Comandos.

batch-suspend-user

O código de exemplo a seguir mostra como usar `batch-suspend-user`.

AWS CLI

Para suspender vários usuários

O `batch-suspend-user` exemplo a seguir suspende os usuários listados da conta especificada do Amazon Chime.

```
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"
```

Saída:

```
{  
  "UserErrors": []  
}
```

- Para API obter detalhes, consulte [BatchSuspendUser](#) na Referência de AWS CLI Comandos.

batch-unsuspend-user

O código de exemplo a seguir mostra como usar `batch-unsuspend-user`.

AWS CLI

Para cancelar a suspensão de vários usuários

O `batch-unsuspend-user` exemplo a seguir remove qualquer suspensão anterior para os usuários listados na conta especificada do Amazon Chime.

```
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"
```

Saída:

```
{  
  "UserErrors": []  
}
```

- Para API obter detalhes, consulte [BatchUnsuspendUser](#) na Referência de AWS CLI Comandos.

batch-update-phone-number

O código de exemplo a seguir mostra como usar `batch-update-phone-number`.

AWS CLI

Para atualizar vários tipos de produtos com números de telefone ao mesmo tempo

O `batch-update-phone-number` exemplo a seguir atualiza os tipos de produtos para todos os números de telefone especificados.

```
aws chime batch-update-phone-number \  
  --update-phone-number-request-items PhoneNumberId=  
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=  
%2B12065550101,ProductType=BusinessCalling
```

Saída:

```
{
  "PhoneNumberErrors": []
}
```

Para atualizar vários números de telefone chamando nomes ao mesmo tempo

O `batch-update-phone-number` exemplo a seguir atualiza os nomes de chamada de todos os números de telefone especificados.

```
aws chime batch-update-phone-number \
  --update-phone-number-request-items PhoneNumberId=
%2B14013143874,CallingName=phonenum1 PhoneNumberId=
%2B14013144061,CallingName=phonenum2
```

Saída:

```
{
  "PhoneNumberErrors": []
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [BatchUpdatePhoneNumber](#) na Referência de AWS CLI Comandos.

batch-update-user

O código de exemplo a seguir mostra como usar `batch-update-user`.

AWS CLI

Para atualizar vários usuários em um único comando

O `batch-update-user` exemplo a seguir atualiza o `LicenseType` para cada um dos usuários listados na conta especificada do Amazon Chime.

```
aws chime batch-update-user \
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

```
--update-user-request-items "UserId=a1b2c3d4-5678-90ab-  
cdef-2222EXAMPLE,LicenseType=Basic" "UserId=a1b2c3d4-5678-90ab-  
cdef-3333EXAMPLE,LicenseType=Basic"
```

Saída:

```
{  
  "UserErrors": []  
}
```

- Para API obter detalhes, consulte [BatchUpdateUser](#) na Referência de AWS CLI Comandos.

create-account

O código de exemplo a seguir mostra como usar create-account.

AWS CLI

Para criar uma conta

O create-account exemplo a seguir cria uma conta do Amazon Chime na conta do AWS administrador.

```
aws chime create-account \  
  --name MyChimeAccount
```

Saída:

```
{  
  "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"
      },
    ]
  }
}

```

Para obter mais informações, consulte [Conceitos básicos](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [CreateAccount](#) na Referência de AWS CLI Comandos.

create-bot

O código de exemplo a seguir mostra como usar `create-bot`.

AWS CLI

Para criar um bot do Amazon Chime

O `create-bot` exemplo a seguir cria um bot para a conta específica do Amazon Chime Enterprise.

```

aws chime create-bot \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --display-name "myBot" \
  --domain "example.com"

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Integrar um bot de bate-papo com o Amazon Chime no Guia](#) do desenvolvedor do Amazon Chime.

- Para API obter detalhes, consulte [CreateBot](#) na Referência de AWS CLI Comandos.

create-phone-number-order

O código de exemplo a seguir mostra como usar `create-phone-number-order`.

AWS CLI

Para criar um pedido de número de telefone

O `create-phone-number-order` exemplo a seguir cria uma ordem de número de telefone para os números de telefone especificados.

```
aws chime create-phone-number-order \  
  --product-type VoiceConnector \  
  --e164-phone-numbers " +12065550100" "+12065550101" "+12065550102"
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [CreatePhoneNumberOrder](#) na Referência de AWS CLI Comandos.

create-proxy-session

O código de exemplo a seguir mostra como usar `create-proxy-session`.

AWS CLI

Para criar uma sessão de proxy

O `create-proxy-session` exemplo a seguir cria uma sessão de proxy com voz e SMS recursos.

```

aws chime create-proxy-session \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --participant-phone-numbers " +14015550101" "+12065550100" \
  --capabilities "Voice" "SMS"

```

Saída:

```

{
  "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"
    }
]
}
}

```

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [CreateProxySession](#) na Referência de AWS CLI Comandos.

create-room-membership

O código de exemplo a seguir mostra como usar `create-room-membership`.

AWS CLI

Para criar uma associação de sala

O `create-room-membership` exemplo a seguir adiciona o usuário especificado à sala de bate-papo como membro da sala de bate-papo.

```

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

```

Saída:

```

{
  "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"
}
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [CreateRoomMembership](#) na Referência de AWS CLI Comandos.

create-room

O código de exemplo a seguir mostra como usar `create-room`.

AWS CLI

Para criar uma sala de bate-papo

O `create-room` exemplo a seguir cria uma sala de bate-papo para a conta especificada do Amazon Chime.

```
aws chime create-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --name chatRoom
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [CreateRoom](#) na Referência de AWS CLI Comandos.

create-user

O código de exemplo a seguir mostra como usar `create-user`.

AWS CLI

Para criar um perfil de usuário para um dispositivo compartilhado

O `create-user` exemplo a seguir cria um perfil de dispositivo compartilhado para o endereço de e-mail especificado.

```
aws chime create-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --email roomdevice@example.com \  
  --user-type SharedDevice
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

Para obter mais informações, consulte [Preparação para configuração](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

create-voice-connector-group

O código de exemplo a seguir mostra como usar `create-voice-connector-group`.

AWS CLI

Para criar um grupo do Amazon Chime Voice Connector

O `create-voice-connector-group` exemplo a seguir cria um grupo de Amazon Chime Voice Connector que inclui o Amazon Chime Voice Connector especificado.

```
aws chime create-voice-connector-group \  
  --name myGroup \  
  --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=2
```

Saída:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "myGroup",  
    "VoiceConnectorItems": [],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de conectores de voz do Amazon Chime no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [CreateVoiceConnectorGroup](#) na Referência de AWS CLI Comandos.

create-voice-connector

O código de exemplo a seguir mostra como usar `create-voice-connector`.

AWS CLI

Para criar um Amazon Chime Voice Connector

O `create-voice-connector` exemplo a seguir cria um Amazon Chime Voice Connector na AWS região especificada, com a criptografia ativada.

```
aws chime create-voice-connector \  
  --name newVoiceConnector \  
  --aws-region us-west-2 \  
  --require-encryption
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [CreateVoiceConnector](#) na Referência de AWS CLI Comandos.

delete-account

O código de exemplo a seguir mostra como usar `delete-account`.

AWS CLI

Para excluir uma conta

O `delete-account` exemplo a seguir exclui a conta especificada.

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo sua conta no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteAccount](#) na Referência de AWS CLI Comandos.

delete-phone-number

O código de exemplo a seguir mostra como usar `delete-phone-number`.

AWS CLI

Para excluir um número de telefone

O `delete-phone-number` exemplo a seguir move o número de telefone especificado para a fila de exclusão.

```
aws chime delete-phone-number \  
  --phone-number-id "+12065550100"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeletePhoneNumber](#) na Referência de AWS CLI Comandos.

delete-proxy-session

O código de exemplo a seguir mostra como usar `delete-proxy-session`.

AWS CLI

Para excluir uma sessão de proxy

O `delete-proxy-session` exemplo a seguir exclui a sessão de proxy especificada.

```
aws chime delete-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk56789l
```

Este comando não produz saída.

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [DeleteProxySession](#) na Referência de AWS CLI Comandos.

delete-room-membership

O código de exemplo a seguir mostra como usar `delete-room-membership`.

AWS CLI

Para remover um usuário como membro de uma sala de bate-papo

O `delete-room-membership` exemplo a seguir remove o membro especificado da sala de bate-papo especificada.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [DeleteRoomMembership](#) na Referência de AWS CLI Comandos.

delete-room

O código de exemplo a seguir mostra como usar `delete-room`.

AWS CLI

Para excluir uma sala de bate-papo

O `delete-room` exemplo a seguir exclui a sala de bate-papo especificada e remove as associações da sala de bate-papo.

```
aws chime delete-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

```
--room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [DeleteRoom](#) na Referência de AWS CLI Comandos.

delete-voice-connector-group

O código de exemplo a seguir mostra como usar `delete-voice-connector-group`.

AWS CLI

title

O `delete-voice-connector-group` exemplo a seguir exclui o grupo especificado do Amazon Chime Voice Connector.

```
aws chime delete-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com grupos de conectores de voz do Amazon Chime no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteVoiceConnectorGroup](#) na Referência de AWS CLI Comandos.

delete-voice-connector-origination

O código de exemplo a seguir mostra como usar `delete-voice-connector-origination`.

AWS CLI

Para excluir as configurações de originação

O `delete-voice-connector-origination` exemplo a seguir exclui o host de origem, a porta, o protocolo, a prioridade e o peso do Amazon Chime Voice Connector especificado.


```
aws chime delete-voice-connector-origination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteVoiceConnectorOrigination](#) na Referência de AWS CLI Comandos.

delete-voice-connector-proxy

O código de exemplo a seguir mostra como usar `delete-voice-connector-proxy`.

AWS CLI

Para excluir uma configuração de proxy

O `delete-voice-connector-proxy` exemplo a seguir exclui a configuração de proxy do seu Amazon Chime Voice Connector.

```
aws chime delete-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída.

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [DeleteVoiceConnectorProxy](#) na Referência de AWS CLI Comandos.

delete-voice-connector-streaming-configuration

O código de exemplo a seguir mostra como usar `delete-voice-connector-streaming-configuration`.

AWS CLI

Para excluir uma configuração de streaming

O `delete-voice-connector-streaming-configuration` exemplo a seguir exclui a configuração de streaming para o Amazon Chime Voice Connector especificado.

```
aws chime delete-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída.

Para obter mais informações, consulte [Streaming de dados do Amazon Chime Voice Connector para o Kinesis](#) no Guia de Administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteVoiceConnectorStreamingConfiguration](#) na Referência de AWS CLI Comandos.

delete-voice-connector-termination-credentials

O código de exemplo a seguir mostra como usar `delete-voice-connector-termination-credentials`.

AWS CLI

Para excluir as credenciais de rescisão

O `delete-voice-connector-termination-credentials` exemplo a seguir exclui as credenciais de rescisão do nome de usuário especificado e do Amazon Chime Voice Connector.

```
aws chime delete-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --usernames "jdoe"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteVoiceConnectorTerminationCredentials](#) na Referência de AWS CLI Comandos.

delete-voice-connector-termination

O código de exemplo a seguir mostra como usar `delete-voice-connector-termination`.

AWS CLI

Para excluir as configurações de rescisão

O `delete-voice-connector-termination` exemplo a seguir exclui as configurações de terminação do Amazon Chime Voice Connector especificado.

```
aws chime delete-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteVoiceConnectorTermination](#) na Referência de AWS CLI Comandos.

`delete-voice-connector`

O código de exemplo a seguir mostra como usar `delete-voice-connector`.

AWS CLI

Para excluir um conector de voz do Amazon Chime

O `delete-voice-connector` exemplo a seguir faz isso

```
aws chime delete-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DeleteVoiceConnector](#) na Referência de AWS CLI Comandos.

`disassociate-phone-number-from-user`

O código de exemplo a seguir mostra como usar `disassociate-phone-number-from-user`.

AWS CLI

Para desassociar um número de telefone de um usuário

O `disassociate-phone-number-from-user` exemplo a seguir dissocia um número de telefone do usuário especificado.

```
aws chime disassociate-phone-number-from-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de números de telefone de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DisassociatePhoneNumberFromUser](#) na Referência de AWS CLI Comandos.

disassociate-phone-numbers-from-voice-connector-group

O código de exemplo a seguir mostra como usar `disassociate-phone-numbers-from-voice-connector-group`.

AWS CLI

Para desassociar números de telefone de um grupo do Amazon Chime Voice Connector

O `disassociate-phone-numbers-from-voice-connector-group` exemplo a seguir desassocia os números de telefone especificados de um grupo do Amazon Chime Voice Connector.

```
aws chime disassociate-phone-numbers-from-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --e164-phone-numbers "+12065550100" "+12065550101"
```

Saída:

```
{  
  "PhoneNumberErrors": []
```

```
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de conectores de voz do Amazon Chime no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [DisassociatePhoneNumbersFromVoiceConnectorGroup](#)na Referência de AWS CLI Comandos.

disassociate-phone-numbers-from-voice-connector

O código de exemplo a seguir mostra como usar `disassociate-phone-numbers-from-voice-connector`.

AWS CLI

Para desassociar números de telefone de um Amazon Chime Voice Connector

O `disassociate-phone-numbers-from-voice-connector` exemplo a seguir desassocia os números de telefone especificados de um Amazon Chime Voice Connector.

```
aws chime disassociate-phone-numbers-from-voice-connector \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --e164-phone-numbers " +12065550100" "+12065550101"
```

Saída:

```
{
  "PhoneNumberErrors": []
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DisassociatePhoneNumbersFromVoiceConnector](#)na Referência de AWS CLI Comandos.

disassociate-signin-delegate-groups-from-account

O código de exemplo a seguir mostra como usar `disassociate-signin-delegate-groups-from-account`.

AWS CLI

Para desassociar grupos de delegados de login

O `disassociate-signin-delegate-groups-from-account` exemplo a seguir desassocia o grupo de delegados de login especificado da conta especificada do Amazon Chime.

```
aws chime disassociate-signin-delegate-groups-from-account \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --group-names "my_users"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de acesso e permissões de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [DisassociateSigninDelegateGroupsFromAccount](#) na Referência de AWS CLI Comandos.

`get-account-settings`

O código de exemplo a seguir mostra como usar `get-account-settings`.

AWS CLI

Para recuperar as configurações de uma conta

O `get-account-settings` exemplo a seguir recupera as configurações da conta especificada.

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Saída:

```
{
  "AccountSettings": {
    "DisableRemoteControl": false,
    "EnableDialOut": false
  }
}
```

Para obter mais informações, consulte [Gerenciando suas contas do Amazon Chime no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetAccountSettings](#) na Referência de AWS CLI Comandos.

get-account

O código de exemplo a seguir mostra como usar get-account.

AWS CLI

Para recuperar os detalhes de uma conta

O get-account exemplo a seguir recupera os detalhes da conta especificada do Amazon Chime.

```
aws chime get-account \  
  --account-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

Saída:

```
{  
  "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"  
      },  
    ]  
  }  
}
```

Para obter mais informações, consulte [Gerenciando suas contas do Amazon Chime no Guia de administração do Amazon Chime](#).

- Para API obter detalhes, consulte [GetAccount](#) na Referência de AWS CLI Comandos.

get-bot

O código de exemplo a seguir mostra como usar `get-bot`.

AWS CLI

Para recuperar detalhes sobre um bot

O `get-bot` exemplo a seguir exibe os detalhes do bot especificado.

```
aws chime get-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Atualizar bots de bate-papo](#) no Guia do desenvolvedor do Amazon Chime.

- Para API obter detalhes, consulte [GetBot](#) na Referência de AWS CLI Comandos.

get-global-settings

O código de exemplo a seguir mostra como usar `get-global-settings`.

AWS CLI

Para obter configurações globais

O `get-global-settings` exemplo a seguir recupera os nomes de bucket do S3 usados para armazenar registros de detalhes de chamadas para Amazon Chime Business Calling e Amazon Chime Voice Connectors associados à conta do administrador. AWS

```
aws chime get-global-settings
```

Saída:

```
{
  "BusinessCalling": {
    "CdrBucket": "s3bucket"
  },
  "VoiceConnector": {
    "CdrBucket": "s3bucket"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de configurações globais](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetGlobalSettings](#) na Referência de AWS CLI Comandos.

get-phone-number-order

O código de exemplo a seguir mostra como usar `get-phone-number-order`.

AWS CLI

Para obter detalhes de um pedido de número de telefone

O `get-phone-number-order` exemplo a seguir exibe os detalhes do pedido de número de telefone especificado.

```
aws chime get-phone-number-order \
  --phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetPhoneNumberOrder](#) na Referência de AWS CLI Comandos.

get-phone-number-settings

O código de exemplo a seguir mostra como usar `get-phone-number-settings`.

AWS CLI

Para recuperar um nome de chamada de saída

O `get-phone-number-settings` exemplo a seguir recupera o nome padrão da chamada de saída para a conta do AWS usuário chamador.

```
aws chime get-phone-number-settings
```

Este comando não produz saída. Saída:

```
{
  "CallingName": "myName",
  "CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetPhoneNumberSettings](#) na Referência de AWS CLI Comandos.

get-phone-number

O código de exemplo a seguir mostra como usar `get-phone-number`.

AWS CLI

Para obter detalhes do número de telefone

O `get-phone-number` exemplo a seguir exibe os detalhes do número de telefone especificado.

```
aws chime get-phone-number \
  --phone-number-id +12065550100
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetPhoneNumber](#) na Referência de AWS CLI Comandos.

get-proxy-session

O código de exemplo a seguir mostra como usar `get-proxy-session`.

AWS CLI

Para obter detalhes da sessão de proxy

O `get-proxy-session` exemplo a seguir lista os detalhes da sessão de proxy especificada.

```

aws chime get-proxy-session \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891

```

Saída:

```

{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [GetProxySession](#) na Referência de AWS CLI Comandos.

get-room

O código de exemplo a seguir mostra como usar `get-room`.

AWS CLI

Para obter os detalhes sobre uma sala de bate-papo

O `get-room` exemplo a seguir exibe detalhes sobre a sala de bate-papo especificada.

```

aws chime get-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j

```

Saída:

```

{
  "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"
}
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [GetRoom](#) na Referência de AWS CLI Comandos.

get-user-settings

O código de exemplo a seguir mostra como usar `get-user-settings`.

AWS CLI

Para recuperar as configurações do usuário

O `get-user-settings` exemplo a seguir exibe as configurações de usuário especificadas.

```
aws chime get-user-settings \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Saída:

```
{
  "UserSettings": {
    "Telephony": {
      "InboundCalling": true,
      "OutboundCalling": true,
      "SMS": true
    }
  }
}
```

Para obter mais informações, consulte [Gerenciamento de números de telefone de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetUserSettings](#) na Referência de AWS CLI Comandos.

get-user

O código de exemplo a seguir mostra como usar `get-user`.

AWS CLI

Para obter detalhes sobre um usuário

O `get-user` exemplo a seguir recupera os detalhes do usuário especificado.

```
aws chime get-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Gerenciamento de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetUser](#) na Referência de AWS CLI Comandos.

get-voice-connector-group

O código de exemplo a seguir mostra como usar `get-voice-connector-group`.

AWS CLI

Para obter detalhes de um grupo do Amazon Chime Voice Connector

O `get-voice-connector-group` exemplo a seguir exibe detalhes do grupo especificado do Amazon Chime Voice Connector.

```
aws chime get-voice-connector-group \
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

Saída:

```
{
  "VoiceConnectorGroup": {
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",
    "Name": "myGroup",
    "VoiceConnectorItems": [],
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
  }
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de conectores de voz do Amazon Chime no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnectorGroup](#) na Referência de AWS CLI Comandos.

`get-voice-connector-logging-configuration`

O código de exemplo a seguir mostra como usar `get-voice-connector-logging-configuration`.

AWS CLI

Para obter detalhes da configuração de registro

O `get-voice-connector-logging-configuration` exemplo a seguir recupera os detalhes da configuração de registro para o Amazon Chime Voice Connector especificado.

```
aws chime get-voice-connector-logging-configuration \
```



```
--voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{
  "LoggingConfiguration": {
    "EnableSIPLogs": true
  }
}
```

Para obter mais informações, consulte [Streaming de mídia do Amazon Chime Voice Connector para o Kinesis](#) no Guia de Administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnectorLoggingConfiguration](#) na Referência de AWS CLI Comandos.

get-voice-connector-origination

O código de exemplo a seguir mostra como usar `get-voice-connector-origination`.

AWS CLI

Para recuperar as configurações de originação

O `get-voice-connector-origination` exemplo a seguir recupera o host de origem, a porta, o protocolo, a prioridade e o peso do Amazon Chime Voice Connector especificado.

```
aws chime get-voice-connector-origination \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{
  "Origination": {
    "Routes": [
      {
        "Host": "10.24.34.0",
        "Port": 1234,
        "Protocol": "TCP",
        "Priority": 1,
        "Weight": 5
      }
    ]
  }
}
```

```
    ],  
    "Disabled": false  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnectorOrigination](#) na Referência de AWS CLI Comandos.

get-voice-connector-proxy

O código de exemplo a seguir mostra como usar `get-voice-connector-proxy`.

AWS CLI

Para obter detalhes da configuração do proxy

O `get-voice-connector-proxy` exemplo a seguir obtém os detalhes da configuração de proxy do seu Amazon Chime Voice Connector.

```
aws chime get-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{  
  "Proxy": {  
    "DefaultSessionExpiryMinutes": 60,  
    "Disabled": false,  
    "PhoneNumberCountries": [  
      "US"  
    ]  
  }  
}
```

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [GetVoiceConnectorProxy](#) na Referência de AWS CLI Comandos.

get-voice-connector-streaming-configuration

O código de exemplo a seguir mostra como usar `get-voice-connector-streaming-configuration`.

AWS CLI

Para obter detalhes da configuração de streaming

O `get-voice-connector-streaming-configuration` exemplo a seguir obtém os detalhes da configuração de streaming para o Amazon Chime Voice Connector especificado.

```
aws chime get-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{  
  "StreamingConfiguration": {  
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

Para obter mais informações, consulte [Streaming de dados do Amazon Chime Voice Connector para o Kinesis](#) no Guia de Administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnectorStreamingConfiguration](#) na Referência de AWS CLI Comandos.

get-voice-connector-termination-health

O código de exemplo a seguir mostra como usar `get-voice-connector-termination-health`.

AWS CLI

Para recuperar detalhes de integridade da rescisão

O `get-voice-connector-termination-health` exemplo a seguir recupera os detalhes de saúde da terminação do Amazon Chime Voice Connector especificado.

```
aws chime get-voice-connector-termination-health \  
--voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{  
  "TerminationHealth": {  
    "Timestamp": "Fri Aug 23 16:45:55 UTC 2019",  
    "Source": "10.24.34.0"  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnectorTerminationHealth](#) na Referência de AWS CLI Comandos.

get-voice-connector-termination

O código de exemplo a seguir mostra como usar `get-voice-connector-termination`.

AWS CLI

Para recuperar as configurações de rescisão

O `get-voice-connector-termination` exemplo a seguir recupera as configurações de terminação do Amazon Chime Voice Connector especificado.

```
aws chime get-voice-connector-termination \  
--voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída. Saída:

```
{  
  "Termination": {  
    "CpsLimit": 1,  
    "DefaultPhoneNumber": "+12065550100",  
    "CallingRegions": [  
      "US"  
    ]  
  }  
}
```

```
    ],
    "CidrAllowedList": [
      "10.24.34.0/23"
    ],
    "Disabled": false
  }
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnectorTermination](#) na Referência de AWS CLI Comandos.

get-voice-connector

O código de exemplo a seguir mostra como usar `get-voice-connector`.

AWS CLI

Para obter detalhes de um Amazon Chime Voice Connector

O `get-voice-connector` exemplo a seguir exibe os detalhes do Amazon Chime Voice Connector especificado.

```
aws chime get-voice-connector \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [GetVoiceConnector](#) na Referência de AWS CLI Comandos.

invite-users

O código de exemplo a seguir mostra como usar `invite-users`.

AWS CLI

Para convidar usuários a participarem do Amazon Chime

O `invite-users` exemplo a seguir envia um e-mail para convidar um usuário para a conta especificada do Amazon Chime.

```
aws chime invite-users \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-email-list "alejandr@example.com" "janed@example.com"
```

Saída:

```
{  
  "Invites": [  
    {  
      "InviteId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
      "Status": "Pending",  
      "EmailAddress": "alejandr@example.com",  
      "EmailStatus": "Sent"  
    },  
    {  
      "InviteId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
      "Status": "Pending",  
      "EmailAddress": "janed@example.com",  
      "EmailStatus": "Sent"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Convidar e suspender usuários no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [InviteUsers](#) na Referência de AWS CLI Comandos.

list-accounts

O código de exemplo a seguir mostra como usar `list-accounts`.

AWS CLI

Para obter uma lista de contas

O `list-accounts` exemplo a seguir recupera uma lista das contas do Amazon Chime na conta do administrador AWS .

```
aws chime list-accounts
```

Saída:

```
{
  "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"
        },
      ]
    }
  ]
}

```

Para obter mais informações, consulte [Gerenciando suas contas do Amazon Chime no Guia de administração do Amazon Chime](#).

- Para API obter detalhes, consulte [ListAccounts](#) na Referência de AWS CLI Comandos.

list-bots

O código de exemplo a seguir mostra como usar `list-bots`.

AWS CLI

Para recuperar uma lista de bots

O `list-bots` exemplo a seguir lista os bots associados à conta especificada do Amazon Chime Enterprise.

```
aws chime list-bots \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45
```

Saída:

```

{
  "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"
  }
}

```



```
}  
}
```

Para obter mais informações, consulte [Usar bots de bate-papo com o Amazon Chime no Guia do desenvolvedor do Amazon Chime](#).

- Para API obter detalhes, consulte [ListBots](#) na Referência de AWS CLI Comandos.

list-phone-number-orders

O código de exemplo a seguir mostra como usar `list-phone-number-orders`.

AWS CLI

Para listar pedidos de números de telefone

O `list-phone-number-orders` exemplo a seguir lista os pedidos com números de telefone associados à conta do administrador do Amazon Chime.

```
aws chime list-phone-number-orders
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [ListPhoneNumberOrders](#) na Referência de AWS CLI Comandos.

list-phone-numbers

O código de exemplo a seguir mostra como usar `list-phone-numbers`.

AWS CLI

Para listar números de telefone para uma conta do Amazon Chime

O `list-phone-numbers` exemplo a seguir lista os números de telefone associados à conta Amazon Chime do administrador.

aws chime list-phone-numbers

Este comando não produz saída. Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [ListPhoneNumbers](#) na Referência de AWS CLI Comandos.

list-proxy-sessions

O código de exemplo a seguir mostra como usar `list-proxy-sessions`.

AWS CLI

Para listar sessões de proxy

O `list-proxy-sessions` exemplo a seguir lista as sessões de proxy do seu Amazon Chime Voice Connector.

```
aws chime list-proxy-sessions \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Saída:

```
{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [ListProxySessions](#) na Referência de AWS CLI Comandos.

list-room-memberships

O código de exemplo a seguir mostra como usar `list-room-memberships`.

AWS CLI

Para listar associações de salas

O `list-room-memberships` exemplo a seguir exibe uma lista dos detalhes da associação para a sala de bate-papo especificada.

```

aws chime list-room-memberships \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j

```

Saída:

```
{
```

```

"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"
  }
]
}

```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [ListRoomMemberships](#) na Referência de AWS CLI Comandos.

list-rooms

O código de exemplo a seguir mostra como usar `list-rooms`.

AWS CLI

Para listar salas de bate-papo

O `list-rooms` exemplo a seguir exibe uma lista de salas de bate-papo na conta especificada. A lista é filtrada somente para as salas de bate-papo às quais o membro especificado pertence.

```
aws chime list-rooms \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [ListRooms](#) na Referência de AWS CLI Comandos.

list-users

O código de exemplo a seguir mostra como usar `list-users`.

AWS CLI

Para listar os usuários em uma conta

O `list-users` exemplo a seguir lista os usuários da conta especificada do Amazon Chime.

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Saída:

```
{  
  "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
    }
  ]
}

```

Para obter mais informações, consulte [Gerenciamento de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [ListUsers](#) na Referência de AWS CLI Comandos.

list-voice-connector-groups

O código de exemplo a seguir mostra como usar `list-voice-connector-groups`.

AWS CLI

Para listar grupos do Amazon Chime Voice Connector para uma conta do Amazon Chime

O `list-voice-connector-groups` exemplo a seguir lista os grupos do Amazon Chime Voice Connector associados à conta Amazon Chime do administrador.

```
aws chime list-voice-connector-groups
```

Saída:

```

{
  "VoiceConnectorGroups": [
    {
      "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl8901",
      "Name": "myGroup",
      "VoiceConnectorItems": [],
      "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
      "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
    }
  ]
}

```

Para obter mais informações, consulte Como [trabalhar com grupos do Amazon Chime Voice Connector no Guia](#) de Administração do Amazon Chime.

- Para API obter detalhes, consulte [ListVoiceConnectorGroups](#) na Referência de AWS CLI Comandos.

list-voice-connector-termination-credentials

O código de exemplo a seguir mostra como usar `list-voice-connector-termination-credentials`.

AWS CLI

Para recuperar uma lista de credenciais de rescisão

O `list-voice-connector-termination-credentials` exemplo a seguir recupera uma lista das credenciais de rescisão do Amazon Chime Voice Connector especificado.

```
aws chime list-voice-connector-termination-credentials \  
--voice-connector-id abcdef1ghij2klmno3pqr4
```

Este comando não produz saída. Saída:

```
{  
  "Usernames": [  
    "jdoe"  
  ]  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [ListVoiceConnectorTerminationCredentials](#) na Referência de AWS CLI Comandos.

list-voice-connectors

O código de exemplo a seguir mostra como usar `list-voice-connectors`.

AWS CLI

Para listar Amazon Chime Voice Connectors para uma conta

O `list-voice-connectors` exemplo a seguir lista os Amazon Chime Voice Connectors associados à conta do chamador.

```
aws chime list-voice-connectors
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [ListVoiceConnectors](#) na Referência de AWS CLI Comandos.

logout-user

O código de exemplo a seguir mostra como usar `logout-user`.

AWS CLI

Para desconectar um usuário

O `logout-user` exemplo a seguir desconecta o usuário especificado.

```
aws chime logout-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Este comando não produz saída.

- Para API obter detalhes, consulte [LogoutUser](#) na Referência de AWS CLI Comandos.

put-voice-connector-logging-configuration

O código de exemplo a seguir mostra como usar `put-voice-connector-logging-configuration`.

AWS CLI

Para adicionar uma configuração de registro para um Amazon Chime Voice Connector

O `put-voice-connector-logging-configuration` exemplo a seguir ativa a configuração de SIP registro para o Amazon Chime Voice Connector especificado.

```
aws chime put-voice-connector-logging-configuration \  
  --voice-connector-id abcdefghijklmno3pqr4 \  
  --logging-configuration EnableSIPLogs=true
```

Saída:

```
{  
  "LoggingConfiguration": {  
    "EnableSIPLogs": true  
  }  
}
```

Para obter mais informações, consulte [Streaming de mídia do Amazon Chime Voice Connector para o Kinesis](#) no Guia de Administração do Amazon Chime.

- Para API obter detalhes, consulte [PutVoiceConnectorLoggingConfiguration](#) na Referência de AWS CLI Comandos.

put-voice-connector-origination

O código de exemplo a seguir mostra como usar `put-voice-connector-origination`.

AWS CLI

Para definir as configurações de originação

O `put-voice-connector-origination` exemplo a seguir configura o host de origem, a porta, o protocolo, a prioridade e o peso do Amazon Chime Voice Connector especificado.

```
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
```

Saída:

```
{  
  "Origination": {  
    "Routes": [  
      {  
        "Host": "10.24.34.0",  
        "Port": 1234,  
        "Protocol": "TCP",  
        "Priority": 1,  
        "Weight": 5  
      }  
    ],  
    "Disabled": false  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [PutVoiceConnectorOrigination](#) na Referência de AWS CLI Comandos.

put-voice-connector-proxy

O código de exemplo a seguir mostra como usar `put-voice-connector-proxy`.

AWS CLI

Para colocar uma configuração de proxy

O `put-voice-connector-proxy` exemplo a seguir define uma configuração de proxy para seu Amazon Chime Voice Connector.

```
aws chime put-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --default-session-expiry-minutes 60 \  
  --phone-number-pool-countries "US"
```

Saída:

```
{  
  "Proxy": {  
    "DefaultSessionExpiryMinutes": 60,  
    "Disabled": false,  
    "PhoneNumberCountries": [  
      "US"  
    ]  
  }  
}
```

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [PutVoiceConnectorProxy](#) na Referência de AWS CLI Comandos.

put-voice-connector-streaming-configuration

O código de exemplo a seguir mostra como usar `put-voice-connector-streaming-configuration`.

AWS CLI

Para criar uma configuração de streaming

O `put-voice-connector-streaming-configuration` exemplo a seguir cria uma configuração de streaming para o Amazon Chime Voice Connector especificado. Ele permite

o streaming de mídia do Amazon Chime Voice Connector para o Amazon Kinesis e define o período de retenção de dados para 24 horas.

```
aws chime put-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --streaming-configuration DataRetentionInHours=24,Disabled=false
```

Saída:

```
{  
  "StreamingConfiguration": {  
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

Para obter mais informações, consulte [Streaming de dados do Amazon Chime Voice Connector para o Kinesis](#) no Guia de Administração do Amazon Chime.

- Para API obter detalhes, consulte [PutVoiceConnectorStreamingConfiguration](#) na Referência de AWS CLI Comandos.

put-voice-connector-termination-credentials

O código de exemplo a seguir mostra como usar `put-voice-connector-termination-credentials`.

AWS CLI

Para configurar as credenciais de rescisão

O `put-voice-connector-termination-credentials` exemplo a seguir define as credenciais de rescisão para o Amazon Chime Voice Connector especificado.

```
aws chime put-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --credentials Username="jdoe",Password="XXXXXXXX"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [PutVoiceConnectorTerminationCredentials](#) na Referência de AWS CLI Comandos.

put-voice-connector-termination

O código de exemplo a seguir mostra como usar `put-voice-connector-termination`.

AWS CLI

Para definir as configurações de rescisão

O `put-voice-connector-termination` exemplo a seguir define as regiões de chamada e as configurações de terminação de host IP permitidas para o Amazon Chime Voice Connector especificado.

```
aws chime put-voice-connector-termination \  
  --voice-connector-id abcdefghijklmno3pqr4 \  
  --termination CallingRegions="US",CidrAllowedList="10.24.34.0/23",Disabled=false
```

Saída:

```
{  
  "Termination": {  
    "CpsLimit": 0,  
    "CallingRegions": [  
      "US"  
    ],  
    "CidrAllowedList": [  
      "10.24.34.0/23"  
    ],  
    "Disabled": false  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [PutVoiceConnectorTermination](#) na Referência de AWS CLI Comandos.

regenerate-security-token

O código de exemplo a seguir mostra como usar `regenerate-security-token`.

AWS CLI

Para regenerar um token de segurança

O `regenerate-security-token` exemplo a seguir regenera o token de segurança do bot especificado.

```
aws chime regenerate-security-token \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Authenticate Chat Bot Requests](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [RegenerateSecurityToken](#) na Referência de AWS CLI Comandos.

reset-personal-pin

O código de exemplo a seguir mostra como usar `reset-personal-pin`.

AWS CLI

Para redefinir a reunião pessoal de um usuário PIN

O `reset-personal-pin` exemplo a seguir redefine a reunião PIN pessoal do usuário especificado.

```
aws chime reset-personal-pin \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Changing Personal Meeting PINs](#) no Guia de Administração do Amazon Chime.

- Para API obter detalhes, consulte [ResetPersonalPin](#) na Referência de AWS CLI Comandos.

restore-phone-number

O código de exemplo a seguir mostra como usar `restore-phone-number`.

AWS CLI

Para restaurar um número de telefone

O `restore-phone-number` exemplo a seguir restaura o número de telefone especificado da fila de exclusão.

```
aws chime restore-phone-number \  
  --phone-number-id "+12065550100"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [RestorePhoneNumber](#) na Referência de AWS CLI Comandos.

search-available-phone-numbers

O código de exemplo a seguir mostra como usar `search-available-phone-numbers`.

AWS CLI

Para pesquisar números de telefone disponíveis

O `search-available-phone-numbers` exemplo a seguir pesquisa os números de telefone disponíveis por código de área.

```
aws chime search-available-phone-numbers \  
  --area-code "206"
```

Saída:

```
{  
  "E164PhoneNumbers": [  
    "+12065550100",  
    "+12065550101",  
    "+12065550102",  
    "+12065550103",  
    "+12065550104",  
    "+12065550105",  
    "+12065550106",  
    "+12065550107",  
    "+12065550108",  
    "+12065550109",  
  ]  
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [SearchAvailablePhoneNumbers](#) na Referência de AWS CLI Comandos.

update-account-settings

O código de exemplo a seguir mostra como usar `update-account-settings`.

AWS CLI

Para atualizar as configurações da sua conta

O `update-account-settings` exemplo a seguir desativa o controle remoto de telas compartilhadas para a conta especificada do Amazon Chime.

```
aws chime update-account-settings \  
  --account-id "123456789012" \  
  --remote-control-sharing "disabled"
```

```
--account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
--account-settings DisableRemoteControl=true
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateAccountSettings](#) na Referência de AWS CLI Comandos.

update-account

O código de exemplo a seguir mostra como usar update-account.

AWS CLI

Para atualizar uma conta

O update-account exemplo a seguir atualiza o nome da conta especificada.

```
aws chime update-account \  
--account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
--name MyAccountName
```

Saída:

```
{  
  "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"  
      },  
    ]  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Renomear sua conta](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdateAccount](#) na Referência de AWS CLI Comandos.

update-bot

O código de exemplo a seguir mostra como usar update-bot.

AWS CLI

Para atualizar um bot

O update-bot exemplo a seguir atualiza o status do bot especificado para impedir sua execução.

```
aws chime update-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \  
  --disabled
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Atualizar bots de bate-papo](#) no Guia do desenvolvedor do Amazon Chime.

- Para API obter detalhes, consulte [UpdateBotna](#) Referência de AWS CLI Comandos.

update-global-settings

O código de exemplo a seguir mostra como usar `update-global-settings`.

AWS CLI

Para atualizar as configurações globais

O `update-global-settings` exemplo a seguir atualiza o bucket do S3 usado para armazenar registros de detalhes de chamadas para Amazon Chime Business Calling e Amazon Chime Voice Connectors associados à conta do administrador. AWS

```
aws chime update-global-settings \  
  --business-calling CdrBucket="s3bucket" \  
  --voice-connector CdrBucket="s3bucket"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de configurações globais](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdateGlobalSettings](#) na Referência de AWS CLI Comandos.

update-phone-number-settings

O código de exemplo a seguir mostra como usar `update-phone-number-settings`.

AWS CLI

Para atualizar o nome de uma chamada de saída

O `update-phone-number-settings` exemplo a seguir atualiza o nome padrão da chamada de saída para a AWS conta do administrador.

```
aws chime update-phone-number-settings \  
  --calling-name "myName"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdatePhoneNumberSettings](#) na Referência de AWS CLI Comandos.

update-phone-number

O código de exemplo a seguir mostra como usar `update-phone-number`.

AWS CLI

Exemplo 1: Para atualizar o tipo de produto de um número de telefone

O `update-phone-number` exemplo a seguir atualiza o tipo de produto do número de telefone especificado.

```
aws chime update-phone-number \  
  --phone-number-id "+12065550100" \  
  --product-type "BusinessCalling"
```

Saída:

```
{  
  "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"  
  }  
}
```



```
}
}
```

Exemplo 2: Para atualizar o nome da chamada de saída para um número de telefone

O `update-phone-number` exemplo a seguir atualiza o nome da chamada de saída para o número de telefone especificado.

as chime `update-phone-number - phone-number-id "+12065550100" --nome de chamada "número de telefone 2"`

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte Como [trabalhar com números de telefone](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdatePhoneNumber](#) na Referência de AWS CLI Comandos.

update-proxy-session

O código de exemplo a seguir mostra como usar `update-proxy-session`.

AWS CLI

Para atualizar uma sessão de proxy

O `update-proxy-session` exemplo a seguir atualiza os recursos da sessão de proxy.

```
aws chime update-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891 \  
  --capabilities "Voice"
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Proxy Phone Sessions](#) no Amazon Chime Developer Guide.

- Para API obter detalhes, consulte [UpdateProxySession](#) na Referência de AWS CLI Comandos.

update-room-membership

O código de exemplo a seguir mostra como usar `update-room-membership`.

AWS CLI

Para atualizar a associação de uma sala

O `update-room-membership` exemplo a seguir modifica a função do membro da sala de bate-papo especificado para `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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [UpdateRoomMembership](#) na Referência de AWS CLI Comandos.

update-room

O código de exemplo a seguir mostra como usar `update-room`.

AWS CLI

Para atualizar uma sala de bate-papo

O `update-room` exemplo a seguir modifica o nome da sala de bate-papo especificada.

```
aws chime update-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \  
  --name teamRoom
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de uma sala de bate-papo](#) no Guia do usuário do Amazon Chime.

- Para API obter detalhes, consulte [UpdateRoom](#) na Referência de AWS CLI Comandos.

update-user-settings

O código de exemplo a seguir mostra como usar `update-user-settings`.

AWS CLI

Para atualizar as configurações do usuário

O `update-user-settings` exemplo a seguir permite que o usuário especificado faça chamadas de entrada e saída e envie e receba SMS mensagens.

```
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}"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de números de telefone de usuários](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdateUserSettings](#) na Referência de AWS CLI Comandos.

update-user

O código de exemplo a seguir mostra como usar update-user.

AWS CLI

Para atualizar os detalhes do usuário

Este exemplo atualiza os detalhes especificados para o usuário especificado.

Comando:

```
aws chime update-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \  
  --license-type "Basic"
```

Saída:

```
{  
  "User": {  
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"  
  }  
}
```

- Para API obter detalhes, consulte [UpdateUser](#) na Referência de AWS CLI Comandos.

update-voice-connector-group

O código de exemplo a seguir mostra como usar update-voice-connector-group.

AWS CLI

Para atualizar os detalhes de um grupo do Amazon Chime Voice Connector

O `update-voice-connector-group` exemplo a seguir atualiza os detalhes do grupo especificado do Amazon Chime Voice Connector.

```
aws chime update-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --name "newGroupName" \  
  --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=1
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de conectores de voz do Amazon Chime no Guia](#) de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdateVoiceConnectorGroup](#) na Referência de AWS CLI Comandos.

update-voice-connector

O código de exemplo a seguir mostra como usar `update-voice-connector`.

AWS CLI

Para atualizar os detalhes de um Amazon Chime Voice Connector

O `update-voice-connector` exemplo a seguir atualiza o nome do Amazon Chime Voice Connector especificado.

```
aws chime update-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --name newName \  
  --require-encryption
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com conectores de voz do Amazon Chime](#) no Guia de administração do Amazon Chime.

- Para API obter detalhes, consulte [UpdateVoiceConnector](#) na Referência de AWS CLI Comandos.

API Exemplos do Cloud Control usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Cloud Control API.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-resource

O código de exemplo a seguir mostra como usar `create-resource`.

AWS CLI

Para criar um recurso

O `create-resource` exemplo a seguir cria um recurso `AWS::Kinesis::Stream`, chamado `ResourceExample`, com um período de retenção de 168 horas e uma contagem de fragmentos de três.

```
aws cloudcontrol create-resource \  
  --type-name AWS::Kinesis::Stream \  
  --desired-state '{"Name": "ResourceExample", "RetentionPeriodHours":168, \  
  "ShardCount":3}'
```

Saída:

```
{  
  "ProgressEvent": {  
    "EventTime": 1632506656.706,  
    "TypeName": "AWS::Kinesis::Stream",  
    "OperationStatus": "IN_PROGRESS",  
    "Operation": "CREATE",  
    "Identifier": "ResourceExample",  
    "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"  
  }  
}
```

Para obter mais informações, consulte [Como criar um recurso](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [CreateResource](#) na Referência de AWS CLI Comandos.

delete-resource

O código de exemplo a seguir mostra como usar `delete-resource`.

AWS CLI

Para excluir um recurso

O `delete-resource` exemplo a seguir exclui um recurso `AWS::Kinesis::Stream` com o identificador `ResourceExample` da sua conta. AWS

```
aws cloudcontrol delete-resource \  
  --type-name AWS::Kinesis::Stream \  
  --identifier ResourceExample
```

Saída:

```
{  
  "ProgressEvent": {  
    "TypeName": "AWS::Kinesis::Stream",  
    "Identifier": "ResourceExample",  
    "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",  
    "Operation": "DELETE",  
    "OperationStatus": "IN_PROGRESS",  
    "EventTime": 1632950300.14  
  }  
}
```

Para obter mais informações, consulte [Excluir um recurso](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [DeleteResource](#) na Referência de AWS CLI Comandos.

`get-resource-request-status`

O código de exemplo a seguir mostra como usar `get-resource-request-status`.

AWS CLI

Para obter as informações de status de uma solicitação de recurso

O `get-resource-request-status` exemplo a seguir retorna informações de status sobre a solicitação de recurso especificada.

```
aws cloudcontrol get-resource-request-status \  
  --request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

Saída:

```
{
  "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"
  }
}
```

Para mais informações, consulte [Gerenciamento de solicitações de operação de recursos](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [GetResourceRequestStatus](#) na Referência de AWS CLI Comandos.

get-resource

O código de exemplo a seguir mostra como usar `get-resource`.

AWS CLI

Para obter o estado atual de um recurso

O `get-resource` exemplo a seguir retorna o estado atual do recurso `AWS::Kinesis::Stream` chamado. `ResourceExample`

```
aws cloudcontrol get-resource \
  --type-name AWS::Kinesis::Stream \
  --identifier ResourceExample
```

Saída:

```
{
  "TypeName": "AWS::Kinesis::Stream",
```

```

    "ResourceDescription": {
      "Identifier": "ResourceExample",
      "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/ResourceExample\",\"RetentionPeriodHours\":168,\"Name\":\"ResourceExample\",
    \"ShardCount\":3}"
    }
  }
}

```

Para obter mais informações, consulte [Lendo o estado atual de um recurso](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [GetResource](#) na Referência de AWS CLI Comandos.

list-resource-requests

O código de exemplo a seguir mostra como usar `list-resource-requests`.

AWS CLI

Para listar as solicitações de operação de recursos ativos

O `list-resource-requests` exemplo a seguir lista as solicitações de recursos CREATE e UPDATE as operações que falharam em sua AWS conta.

```

aws cloudcontrol list-resource-requests \
  --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED

```

Saída:

```

{
  "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"
    }
  ]
}

```

```
]
}
```

Para mais informações, consulte [Gerenciamento de solicitações de operação de recursos](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [ListResourceRequests](#) na Referência de AWS CLI Comandos.

list-resources

O código de exemplo a seguir mostra como usar `list-resources`.

AWS CLI

Para listar os recursos de um determinado tipo

O `list-resources` exemplo a seguir lista os recursos AWS::Kinesis::Stream provisionados em sua conta. AWS

```
aws cloudcontrol list-resources \
  --type-name AWS::Kinesis::Stream
```

Saída:

```
{
  "TypeName": "AWS::Kinesis::Stream",
  "ResourceDescriptions": [
    {
      "Identifier": "MyKinesisStream",
      "Properties": "{\"Name\":\"MyKinesisStream\"}"
    },
    {
      "Identifier": "AnotherStream",
      "Properties": "{\"Name\":\"AnotherStream\"}"
    }
  ]
}
```

Para obter mais informações, consulte [Descobrendo recursos](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [ListResources](#) na Referência de AWS CLI Comandos.

update-resource

O código de exemplo a seguir mostra como usar `update-resource`.

AWS CLI

Para atualizar as propriedades de um recurso existente

O `update-resource` exemplo a seguir atualiza a política de retenção de um LogGroup recurso AWS: :Logs:: nomeado ExampleLogGroup para 90 dias.

```
aws cloudcontrol update-resource \  
  --type-name AWS::Logs::LogGroup \  
  --identifier ExampleLogGroup \  
  --patch-document "[{\\"op\\":\\"replace\\",\\"path\\":\\"/RetentionInDays\\",\\"value\\":90}]"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Atualização de um recurso](#) no Guia do API usuário do Cloud Control.

- Para API obter detalhes, consulte [UpdateResource](#) na Referência de AWS CLI Comandos.

AWS Cloud Map exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Cloud Map.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-private-dns-namespace

O código de exemplo a seguir mostra como usar `create-private-dns-namespace`.

AWS CLI

Para criar um DNS namespace privado

O `create-private-dns-namespace` exemplo a seguir cria um DNS namespace privado.

```
aws servicediscovery create-private-dns-namespace \  
  --name example.com \  
  --vpc vpc-1c56417b
```

Saída:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"  
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation`. Para obter mais informações, consulte [get-operation](#).

Para obter mais informações, consulte [Criação de namespaces](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [CreatePrivateDnsNamespacena](#) Referência de AWS CLI Comandos.

create-service

O código de exemplo a seguir mostra como usar create-service.

AWS CLI

Para criar um serviço

O create-service exemplo a seguir cria um serviço.

```
aws servicediscovery create-service \  
  --name myservice \  
  --namespace-id ns-ylexjili4cdxy3xm \  
  --dns-config "NamespaceId=ns-  
ylexjili4cdxy3xm,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de serviços](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [CreateService](#) na Referência de AWS CLI Comandos.

delete-namespace

O código de exemplo a seguir mostra como usar `delete-namespace`.

AWS CLI

Para excluir um namespace

O `delete-namespace` exemplo a seguir exclui um namespace.

```
aws servicediscovery delete-namespace \  
  --id ns-ylexjili4cdxy3xm
```

Saída:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6dtk"  
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation`. Para obter mais informações, consulte [get-operation](#).

Para obter mais informações, consulte [Excluindo namespaces](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [DeleteNamespace](#) na Referência de AWS CLI Comandos.

delete-service

O código de exemplo a seguir mostra como usar `delete-service`.

AWS CLI

Como excluir um serviço

O `delete-service` exemplo a seguir exclui um serviço.

```
aws servicediscovery delete-service \  
  --id srv-p5zdwlg5uvvzjita
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo serviços](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [DeleteService](#) na Referência de AWS CLI Comandos.

deregister-instance

O código de exemplo a seguir mostra como usar `deregister-instance`.

AWS CLI

Para cancelar o registro de uma instância de serviço

O `deregister-instance` exemplo a seguir cancela o registro de uma instância de serviço.

```
aws servicediscovery deregister-instance \  
  --service-id srv-p5zdwlg5uvvzjita \  
  --instance-id myservice-53
```

Saída:

```
{  
  "OperationId": "4yejorelbukcjzpnr6t1mrghsjwpngf4-k98rnaiq"  
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation`. Para obter mais informações, consulte [get-operation](#).

Para obter mais informações, consulte [Cancelamento do registro de instâncias de serviço](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [DeregisterInstance](#) na Referência de AWS CLI Comandos.

discover-instances

O código de exemplo a seguir mostra como usar `discover-instances`.

AWS CLI

Para descobrir instâncias registradas

O `discover-instances` exemplo a seguir descobre instâncias registradas.

```
aws servicediscovery discover-instances \  
  --namespace-name example.com \  
  --instance-ids myinstance-1 myinstance-2
```

```
--service-name myservice \  
--max-results 10 \  
--health-status ALL
```

Saída:

```
{  
  "Instances": [  
    {  
      "InstanceId": "myservice-53",  
      "NamespaceName": "example.com",  
      "ServiceName": "myservice",  
      "HealthStatus": "UNKNOWN",  
      "Attributes": {  
        "AWS_INSTANCE_IPV4": "172.2.1.3",  
        "AWS_INSTANCE_PORT": "808"  
      }  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DiscoverInstances](#) na Referência de AWS CLI Comandos.

get-operation

O código de exemplo a seguir mostra como usar `get-operation`.

AWS CLI

Para obter o resultado de uma operação

O `get-operation` exemplo a seguir obtém o resultado de uma operação.

```
aws servicediscovery get-operation \  
--operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

Saída:

```
{  
  "Operation": {  
    "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",  
    "Type": "CREATE_NAMESPACE",
```

```
    "Status": "SUCCESS",
    "CreateDate": 1587055860.121,
    "UpdateDate": 1587055900.469,
    "Targets": {
      "NAMESPACE": "ns-ylexjili4cdxy3xm"
    }
  }
}
```

- Para API obter detalhes, consulte [GetOperation](#) na Referência de AWS CLI Comandos.

list-instances

O código de exemplo a seguir mostra como usar `list-instances`.

AWS CLI

Para listar instâncias de serviço

O `list-instances` exemplo a seguir lista as instâncias de serviço.

```
aws servicediscovery list-instances \
  --service-id srv-qzpwvt2tfqcegapy
```

Saída:

```
{
  "Instances": [
    {
      "Id": "i-06bdabbae60f65a4e",
      "Attributes": {
        "AWS_INSTANCE_IPV4": "172.2.1.3",
        "AWS_INSTANCE_PORT": "808"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Visualização de uma lista de instâncias de serviço](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [ListInstances](#) na Referência de AWS CLI Comandos.

list-namespaces

O código de exemplo a seguir mostra como usar `list-namespaces`.

AWS CLI

Para listar namespaces

O `list-namespaces` exemplo a seguir lista os namespaces.

```
aws servicediscovery list-namespaces
```

Saída:

```
{
  "Namespaces": [
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
a3ccy2e7e3a7rile",
      "CreateDate": 1585354387.357,
      "Id": "ns-a3ccy2e7e3a7rile",
      "Name": "local",
      "Properties": {
        "DnsProperties": {
          "HostedZoneId": "Z06752353VBUDTC32S84S"
        },
        "HttpProperties": {
          "HttpName": "local"
        }
      },
      "Type": "DNS_PRIVATE"
    },
    {
      "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
pocfyjtrsmwtvcxx",
      "CreateDate": 1586468974.698,
      "Description": "My second namespace",
      "Id": "ns-pocfyjtrsmwtvcxx",
      "Name": "My-second-namespace",
      "Properties": {
        "DnsProperties": {},
        "HttpProperties": {
          "HttpName": "My-second-namespace"
        }
      }
    }
  ]
}
```

```

    },
    "Type": "HTTP"
  },
  {
    "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
ylexjili4cdxy3xm",
    "CreateDate": 1587055896.798,
    "Id": "ns-ylexjili4cdxy3xm",
    "Name": "example.com",
    "Properties": {
      "DnsProperties": {
        "HostedZoneId": "Z09983722P0QME1B3KC8I"
      },
      "HttpProperties": {
        "HttpName": "example.com"
      }
    },
    "Type": "DNS_PRIVATE"
  }
]
}

```

Para obter mais informações, consulte [Visualização de uma lista de namespaces](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [ListNamespaces](#) na Referência de AWS CLI Comandos.

list-services

O código de exemplo a seguir mostra como usar `list-services`.

AWS CLI

Para listar serviços

O `list-services` exemplo a seguir lista os serviços.

```
aws servicediscovery list-services
```

Saída:

```
{
  "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
    }
  ]
}

```

Para obter mais informações, consulte [Visualizando uma lista de serviços](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [ListServices](#) na Referência de AWS CLI Comandos.

register-instance

O código de exemplo a seguir mostra como usar `register-instance`.

AWS CLI

Para registrar uma instância de serviço

O `register-instance` exemplo a seguir registra uma instância de serviço.

```

aws servicediscovery register-instance \
  --service-id srv-p5zdwlg5uvvzjita \
  --instance-id myservice-53 \
  --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808

```

Saída:

```

{
  "OperationId": "4yejorelbukcjpnr6t1mrghsjwpngf4-k95yg2u7"
}

```

```
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation`. Para obter mais informações, consulte [get-operation](#).

Para obter mais informações, consulte [Registro de instâncias](#) no AWS Cloud Map Developer Guide.

- Para API obter detalhes, consulte [RegisterInstance](#) na Referência de AWS CLI Comandos.

AWS Cloud9 exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Cloud9.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-environment-ec2

O código de exemplo a seguir mostra como usar `create-environment-ec2`.

AWS CLI

Para criar um ambiente de AWS desenvolvimento EC2 Cloud9

O `create-environment-ec2` exemplo a seguir cria um AWS ambiente de desenvolvimento Cloud9 com as configurações especificadas, inicia uma instância do Amazon Elastic Compute Cloud (EC2Amazon) e, em seguida, se conecta da instância ao 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
```

Saída:

```
{  
  "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"  
}
```

Para obter mais informações, consulte o Guia do usuário da [Criação de um EC2 ambiente](#) no AWS Cloud9.

- Para API obter detalhes, consulte [CreateEnvironmentEc2](#) em Referência de AWS CLI comandos.

create-environment-membership

O código de exemplo a seguir mostra como usar create-environment-membership.

AWS CLI

Para adicionar um membro do ambiente a um ambiente de desenvolvimento do AWS Cloud9

Este exemplo adiciona o membro do ambiente especificado ao ambiente de desenvolvimento do AWS Cloud9 especificado.

Comando:

```
aws cloud9 create-environment-membership --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/  
AnotherDemoUser --permissions read-write
```

Saída:

```
{  
  "membership": {  
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",  
    "userId": "AIDAJ3LOROMOUXTBSUGEX",  
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
```



```
    "permissions": "read-write"  
  }  
}
```

- Para API obter detalhes, consulte [CreateEnvironmentMembership](#) na Referência de AWS CLI Comandos.

delete-environment-membership

O código de exemplo a seguir mostra como usar `delete-environment-membership`.

AWS CLI

Para excluir um membro do ambiente de um ambiente de desenvolvimento do AWS Cloud9

Este exemplo exclui o membro do ambiente especificado do ambiente de desenvolvimento do AWS Cloud9 especificado.

Comando:

```
aws cloud9 delete-environment-membership --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/  
AnotherDemoUser
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DeleteEnvironmentMembership](#) na Referência de AWS CLI Comandos.

delete-environment

O código de exemplo a seguir mostra como usar `delete-environment`.

AWS CLI

Para excluir um ambiente de desenvolvimento do AWS Cloud9

Este exemplo exclui o ambiente de desenvolvimento do AWS Cloud9 especificado. Se uma EC2 instância da Amazon estiver conectada ao ambiente, a instância também será encerrada.

Comando:

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DeleteEnvironment](#) na Referência de AWS CLI Comandos.

describe-environment-memberships

O código de exemplo a seguir mostra como usar `describe-environment-memberships`.

AWS CLI

Para obter informações sobre os membros do ambiente de um ambiente de desenvolvimento AWS Cloud9

Este exemplo obtém informações sobre os membros do ambiente do ambiente de desenvolvimento AWS Cloud9 especificado.

Comando:

```
aws cloud9 describe-environment-memberships --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
    }  
  ]  
}
```

Para obter informações sobre o proprietário de um ambiente de desenvolvimento AWS Cloud9

Este exemplo obtém informações sobre o proprietário do ambiente de desenvolvimento AWS Cloud9 especificado.

Comando:

```
aws cloud9 describe-environment-memberships --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

Saída:

```
{  
  "memberships": [  
    {  
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",  
      "userId": "AIDAJNUEDQAQWFELJDLEX",  
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",  
      "permissions": "owner"  
    }  
  ]  
}
```

Para obter informações sobre um membro do ambiente para vários ambientes de desenvolvimento do AWS Cloud9

Este exemplo obtém informações sobre o membro do ambiente especificado para vários ambientes de desenvolvimento do AWS Cloud9.

Comando:

```
aws cloud9 describe-environment-memberships --user-  
arn arn:aws:iam::123456789012:user/MyDemoUser
```

Saída:

```
{  
  "memberships": [  
    {  
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",  
      "userId": "AIDAJNUEDQAQWFELJDLEX",  
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",  
      "permissions": "owner"  
    }  
  ]  
}
```

```
{
  "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"
}
]
```

- Para API obter detalhes, consulte [DescribeEnvironmentMemberships](#) na Referência de AWS CLI Comandos.

describe-environment-status

O código de exemplo a seguir mostra como usar `describe-environment-status`.

AWS CLI

Para obter informações de status de um ambiente de desenvolvimento AWS Cloud9

Este exemplo obtém informações de status para o ambiente de desenvolvimento AWS Cloud9 especificado.

Comando:

```
aws cloud9 describe-environment-status --environment-  
id 685f892f431b45c2b28cb69eadcdb0EX
```

Saída:

```
{
  "status": "ready",
  "message": "Environment is ready to use"
}
```

- Para API obter detalhes, consulte [DescribeEnvironmentStatus](#) na Referência de AWS CLI Comandos.

describe-environments

O código de exemplo a seguir mostra como usar `describe-environments`.

AWS CLI

Para obter informações sobre os ambientes de desenvolvimento do AWS Cloud9

Este exemplo obtém informações sobre os ambientes de desenvolvimento AWS Cloud9 especificados.

Comando:

```
aws cloud9 describe-environments --environment-ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeEnvironments](#) na Referência de AWS CLI Comandos.

list-environments

O código de exemplo a seguir mostra como usar `list-environments`.

AWS CLI

Para obter uma lista dos identificadores de ambiente de AWS desenvolvimento Cloud9 disponíveis

Este exemplo obtém uma lista dos identificadores de ambiente de AWS desenvolvimento do Cloud9 disponíveis.

Comando:

```
aws cloud9 list-environments
```

Saída:

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- Para API obter detalhes, consulte [ListEnvironments](#) na Referência de AWS CLI Comandos.

update-environment-membership

O código de exemplo a seguir mostra como usar `update-environment-membership`.

AWS CLI

Para alterar as configurações de um membro de ambiente existente para um ambiente de desenvolvimento AWS Cloud9

Este exemplo altera as configurações do membro do ambiente existente especificado para o ambiente de desenvolvimento AWS Cloud9 especificado.

Comando:

```
aws cloud9 update-environment-membership --environment-id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/AnotherDemoUser --permissions read-only
```

Saída:

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUCTBSUGEX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
}
```

- Para API obter detalhes, consulte [UpdateEnvironmentMembership](#) na Referência de AWS CLI Comandos.

update-environment

O código de exemplo a seguir mostra como usar update-environment.

AWS CLI

Para alterar as configurações de um ambiente de desenvolvimento AWS Cloud9 existente

Este exemplo altera as configurações especificadas do ambiente de desenvolvimento AWS Cloud9 existente especificado.

Comando:

```
aws cloud9 update-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX  
--name my-changed-demo-env --description "My changed demonstration development  
environment."
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [UpdateEnvironment](#) na Referência de AWS CLI Comandos.

AWS CloudFormation exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS CloudFormation.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

activate-type

O código de exemplo a seguir mostra como usar `activate-type`.

AWS CLI

Para ativar um tipo

O `activate-type` exemplo a seguir ativa uma extensão pública de terceiros, disponibilizando-a para uso em modelos de pilha.

```
aws cloudformation activate-type \  

```



```
--region us-west-2 \  
--type RESOURCE \  
--type-name Example::Test::1234567890abcdef0 \  
--type-name-alias Example::Test::Alias
```

Saída:

```
{  
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-  
Test-Alias"  
}
```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [ActivateType](#) na Referência de AWS CLI Comandos.

batch-describe-type-configurations

O código de exemplo a seguir mostra como usar batch-describe-type-configurations.

AWS CLI

Para descrever em lote uma configuração de tipo

O batch-describe-type-configurations exemplo a seguir configura os dados para o 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"
```

Saída:

```
{  
  "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\":
        \"examplekey\", \n
        \n        \"ApplicationKey\": \"examplekey1\", \n
        \n        \"ApiURL\": \"exampleurl\"\n
        \n      }\n
        \n    },
        \"LastUpdated\": \"2021-10-01T15:25:46.210000+00:00\",
        \"TypeArn\": \"arn:aws:cloudformation:us-east-1:123456789012:type/resource/
        Example-Test-Type\"
      }
    ]
  }

```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [BatchDescribeTypeConfigurations](#) na Referência de AWS CLI Comandos.

cancel-update-stack

O código de exemplo a seguir mostra como usar `cancel-update-stack`.

AWS CLI

Para cancelar uma atualização de pilha que está em andamento

O comando `cancel-update-stack` a seguir cancela uma atualização de pilha na pilha `myteststack`:

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Para API obter detalhes, consulte [CancelUpdateStack](#) na Referência de AWS CLI Comandos.

continue-update-rollback

O código de exemplo a seguir mostra como usar `continue-update-rollback`.

AWS CLI

Para fazer nova tentativa de reversão de atualização

O exemplo `continue-update-rollback` a seguir retoma uma operação de reversão de uma atualização de pilha anterior em que houve falha.

```
aws cloudformation continue-update-rollback \  
  --stack-name my-stack
```

Este comando não produz saída.

- Para API obter detalhes, consulte [ContinueUpdateRollback](#) na Referência de AWS CLI Comandos.

create-change-set

O código de exemplo a seguir mostra como usar create-change-set.

AWS CLI

Para criar um conjunto de alterações

O create-change-set exemplo a seguir cria um conjunto de alterações com o CAPABILITY_IAM recurso. O arquivo `template.yaml` é um AWS CloudFormation modelo na pasta atual que define uma pilha que inclui IAM recursos.

```
aws cloudformation create-change-set \  
  --stack-name my-application \  
  --change-set-name my-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM
```

Saída:

```
{  
  "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"  
}
```

- Para API obter detalhes, consulte [CreateChangeSet](#) na Referência de AWS CLI Comandos.

create-stack-instances

O código de exemplo a seguir mostra como usar create-stack-instances.

AWS CLI

Para criar instâncias de pilha

O `create-stack-instances` exemplo a seguir cria instâncias de um conjunto de pilhas em duas contas e em quatro regiões. A configuração de tolerância a falhas garante que a atualização seja tentada em todas as contas e regiões, mesmo que algumas pilhas não possam ser criadas.

```
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
```

Saída:

```
{  
  "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"  
}
```

Para criar um conjunto de pilhas, use o `create-stack-set` comando.

- Para API obter detalhes, consulte [CreateStackInstances](#) na Referência de AWS CLI Comandos.

`create-stack-set`

O código de exemplo a seguir mostra como usar `create-stack-set`.

AWS CLI

Para criar um conjunto de pilhas

O `create-stack-set` exemplo a seguir cria um conjunto de pilhas usando o modelo de YAML arquivo especificado. `template.yaml` é um AWS CloudFormation modelo na pasta atual que define uma pilha.

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stack-set \  
  --template-body file://template.yaml \  
  --description "SNS topic"
```

Saída:

```
{
  "StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"
}
```

Para adicionar instâncias de pilha ao conjunto de pilhas, use o `create-stack-instances` comando.

- Para API obter detalhes, consulte [CreateStackSet](#) na Referência de AWS CLI Comandos.

create-stack

O código de exemplo a seguir mostra como usar `create-stack`.

AWS CLI

Para criar uma AWS CloudFormation pilha

O exemplo de `create-stacks` a seguir cria uma pilha com o nome de `myteststack` usando o arquivo de modelo `sampletemplate.json`:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Saída:

```
{
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

Para obter mais informações, consulte [Stacks](#) no Guia do AWS CloudFormation usuário.

- Para API obter detalhes, consulte [CreateStack](#) na Referência de AWS CLI Comandos.

deactivate-type

O código de exemplo a seguir mostra como usar `deactivate-type`.

AWS CLI

Para desativar um tipo

O `deactivate-type` exemplo a seguir desativa uma extensão pública que foi ativada anteriormente nessa conta e região.

```
aws cloudformation deactivate-type \  
  --region us-west-2 \  
  --type MODULE \  
  --type-name Example::Test::Type::MODULE
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [DeactivateType](#) na Referência de AWS CLI Comandos.

`delete-change-set`

O código de exemplo a seguir mostra como usar `delete-change-set`.

AWS CLI

Para excluir um conjunto de alterações

O `delete-change-set` exemplo a seguir exclui um conjunto de alterações especificando o nome do conjunto de alterações e o nome da pilha.

```
aws cloudformation delete-change-set \  
  --stack-name my-stack \  
  --change-set-name my-change-set
```

Este comando não produz saída.

O `delete-change-set` exemplo a seguir exclui um conjunto de alterações especificando a totalidade ARN do conjunto de alterações.

```
aws cloudformation delete-change-set \  
  --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteChangeSet](#) na Referência de AWS CLI Comandos.

delete-stack-instances

O código de exemplo a seguir mostra como usar `delete-stack-instances`.

AWS CLI

Para excluir instâncias de pilha

O `delete-stack-instances` exemplo a seguir exclui instâncias de um conjunto de pilhas em duas contas em duas regiões e encerra as pilhas.

```
aws cloudformation delete-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 567890123456 \  
  --regions us-east-1 us-west-1 \  
  --no-retain-stacks
```

Saída:

```
{  
  "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"  
}
```

Para excluir um conjunto de pilhas vazio, use o `delete-stack-set` comando.

- Para API obter detalhes, consulte [DeleteStackInstances](#) na Referência de AWS CLI Comandos.

delete-stack-set

O código de exemplo a seguir mostra como usar `delete-stack-set`.

AWS CLI

Para excluir um conjunto de pilhas

O comando a seguir exclui o conjunto de pilhas vazio especificado. O conjunto de pilhas deve estar vazio.

```
aws cloudformation delete-stack-set \  
  --stack-set-name my-stack-set
```

Este comando não produz saída.

Para excluir instâncias do conjunto de pilhas, use o `delete-stack-instances` comando.

- Para API obter detalhes, consulte [DeleteStackSet](#) na Referência de AWS CLI Comandos.

delete-stack

O código de exemplo a seguir mostra como usar `delete-stack`.

AWS CLI

Para excluir uma pilha

O exemplo de `delete-stack` a seguir exclui a pilha especificada.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteStack](#) na Referência de AWS CLI Comandos.

deploy

O código de exemplo a seguir mostra como usar `deploy`.

AWS CLI

O comando a seguir implanta o modelo nomeado `template.json` em uma pilha chamada: `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
```

- Para API obter detalhes, consulte [Implantar](#) na Referência de AWS CLI Comandos.

deregister-type

O código de exemplo a seguir mostra como usar `deregister-type`.

AWS CLI

Para cancelar o registro de uma versão de texto

O `deregister-type` exemplo a seguir remove a versão do tipo especificado do uso ativo no CloudFormation registro, para que ela não possa mais ser usada em CloudFormation operações.

```
aws cloudformation deregister-type \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --version-id 00000002
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o CloudFormation Registro](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [DeregisterType](#) na Referência de AWS CLI Comandos.

`describe-account-limits`

O código de exemplo a seguir mostra como usar `describe-account-limits`.

AWS CLI

Para obter informações sobre os limites da sua conta

O comando a seguir recupera uma lista de limites regionais para a conta atual.

```
aws cloudformation describe-account-limits
```

Saída:

```
{  
  "AccountLimits": [  
    {  
      "Name": "StackLimit",  
      "Value": 200  
    },  
    {  
      "Name": "StackOutputsLimit",
```

```

        "Value": 60
      },
      {
        "Name": "ConcurrentResourcesLimit",
        "Value": 2500
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [DescribeAccountLimits](#) na Referência de AWS CLI Comandos.

describe-change-set

O código de exemplo a seguir mostra como usar `describe-change-set`.

AWS CLI

Para obter informações sobre um conjunto de alterações

O `describe-change-set` exemplo a seguir exibe os detalhes do conjunto de alterações especificado pelo nome do conjunto de alterações e nome da pilha.

```

aws cloudformation describe-change-set \
  --change-set-name my-change-set \
  --stack-name my-stack

```

O `describe-change-set` exemplo a seguir exibe os detalhes do conjunto de alterações especificado pela totalidade ARN do conjunto de alterações:

```

aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784

```

Saída:

```

{
  "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
}

```

- Para API obter detalhes, consulte [DescribeChangeSet](#) na Referência de AWS CLI Comandos.

describe-publisher

O código de exemplo a seguir mostra como usar `describe-publisher`.

AWS CLI

Para descrever um editor

O `describe-publisher` exemplo a seguir configura as informações de um editor.

```
aws cloudformation describe-publisher \  
  --region us-west-2 \  
  --publisher-id 000q6TfUovXsEMmgKowxDZLLwqr2QUsh
```

Saída:

```
{  
  "PublisherId": "000q6TfUovXsEMmgKowxDZLLwqr2QUshd2e75c8c",  
  "PublisherStatus": "VERIFIED",  
  "IdentityProvider": "AWS_Marketplace",  
  "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?  
id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"  
}
```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [DescribePublisher](#) na Referência de AWS CLI Comandos.

describe-stack-drift-detection-status

O código de exemplo a seguir mostra como usar `describe-stack-drift-detection-status`.

AWS CLI

Para verificar o status de uma operação de detecção de desvio

O `describe-stack-drift-detection-status` exemplo a seguir exibe o status de uma operação de detecção de desvio. Obtenha o by ID executando o `detect-stack-drift` comando.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-name StackName \  
  --stack-id StackId
```

```
--stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [DescribeStackDriftDetectionStatus](#) na Referência de AWS CLI Comandos.

describe-stack-events

O código de exemplo a seguir mostra como usar `describe-stack-events`.

AWS CLI

Para descrever os eventos da pilha

O exemplo `describe-stack-events` a seguir exibe os dois eventos mais recentes da pilha especificada.

```
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": "eyJJOZXh0VG9XMPLi0iBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQi0iAyfQ=="
}

```

- Para API obter detalhes, consulte [DescribeStackEvents](#) na Referência de AWS CLI Comandos.

describe-stack-instance

O código de exemplo a seguir mostra como usar `describe-stack-instance`.

AWS CLI

Para descrever uma instância de pilha

O comando a seguir descreve uma instância do conjunto de pilhas especificado na conta e na região especificadas. O conjunto de pilhas está na região e na conta atuais, e a instância está na `us-west-2` região da conta `123456789012`.

```

aws cloudformation describe-stack-instance \
  --stack-set-name my-stack-set \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-west-2

```

Saída:

```
{
  "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/
Fdt89BXMPLyiqbMNIjVkJ55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj967lhriaG/dWMobS040o=)."
  }
}
```

- Para API obter detalhes, consulte [DescribeStackInstance](#) na Referência de AWS CLI Comandos.

describe-stack-resource-drifts

O código de exemplo a seguir mostra como usar `describe-stack-resource-drifts`.

AWS CLI

Para obter informações sobre recursos que derivaram da definição da pilha

O comando a seguir exibe informações sobre recursos desviados para a pilha especificada. Para iniciar a detecção de desvios, use o `detect-stack-drift` comando. :

```
aws cloudformation describe-stack-resource-drifts \
  --stack-name my-stack
```

A saída mostra uma função AWS Lambda que foi modificada: `out-of-band`

```
{
  "StackResourceDrifts": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
```

```

        "LogicalResourceId": "function",
        "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
        "ResourceType": "AWS::Lambda::Function",
        "ExpectedProperties": "{\"Description\":\"Write a file to S3.\",
\\\"Environment\\\":{\\\"Variables\\\":{\\\"bucket\\\":\\\"my-stack-bucket-1vc62xmplgguf
\\\"}},\\\"Handler\\\":\\\"index.handler\\\",\\\"MemorySize\\\":128,\\\"Role\\\":
\\\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLE0M9E\\\",\\\"Runtime\\\":
\\\"nodejs10.x\\\",\\\"Tags\\\":[{\\\"Key\\\":\\\"lambda:createdBy\\\",\\\"Value\\\":\\\"SAM\\\"}],\\\"Timeout
\\\":900,\\\"TracingConfig\\\":{\\\"Mode\\\":\\\"Active\\\"}}\",
        "ActualProperties": "{\"Description\":\"Write a file to S3.\",
\\\"Environment\\\":{\\\"Variables\\\":{\\\"bucket\\\":\\\"my-stack-bucket-1vc62xmplgguf
\\\"}},\\\"Handler\\\":\\\"index.handler\\\",\\\"MemorySize\\\":256,\\\"Role\\\":
\\\"arn:aws:iam::123456789012:role/my-functionRole-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"
    }
]
}

```

- Para API obter detalhes, consulte [DescribeStackResourceDrifts](#) na Referência de AWS CLI Comandos.

describe-stack-resource

O código de exemplo a seguir mostra como usar `describe-stack-resource`.

AWS CLI

Para obter informações detalhadas sobre o recurso de uma pilha

O exemplo de `describe-stack-resource` a seguir exibe os detalhes do recurso denominado `MyFunction` da pilha especificada.

```
aws cloudformation describe-stack-resource \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [DescribeStackResource](#) na Referência de AWS CLI Comandos.

`describe-stack-resources`

O código de exemplo a seguir mostra como usar `describe-stack-resources`.

AWS CLI

Para obter informações detalhadas sobre o recurso de uma pilha

O exemplo de `describe-stack-resources` a seguir exibe os detalhes dos recursos na pilha especificada.

```
aws cloudformation describe-stack-resources \  
  --stack-name my-stack
```

Saída:

```
{  
  "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-HIZXMPLEOM9E",  
      "ResourceType": "AWS::IAM::Role",
```

```

        "Timestamp": "2019-10-02T04:34:06.350Z",
        "ResourceStatus": "CREATE_COMPLETE",
        "DriftInformation": {
            "StackResourceDriftStatus": "IN_SYNC"
        }
    }
]
}

```

- Para API obter detalhes, consulte [DescribeStackResources](#) na Referência de AWS CLI Comandos.

describe-stack-set-operation

O código de exemplo a seguir mostra como usar `describe-stack-set-operation`.

AWS CLI

Para obter informações sobre uma operação de conjunto de pilhas

O exemplo `describe-stack-set-operation` a seguir exibe detalhes de uma operação de atualização no conjunto de pilhas especificado.

```

aws cloudformation describe-stack-set-operation \
  --stack-set-name enable-config \
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0

```

Saída:

```

{
  "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"
}
}

```

- Para API obter detalhes, consulte [DescribeStackSetOperation](#) na Referência de AWS CLI Comandos.

describe-stack-set

O código de exemplo a seguir mostra como usar `describe-stack-set`.

AWS CLI

Para obter informações sobre um conjunto de pilhas

O exemplo `describe-stack-set` a seguir exibe detalhes sobre o conjunto de pilhas especificado.

```

aws cloudformation describe-stack-set \
  --stack-set-name my-stack-set

```

Saída:

```

{
  "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"
  }
}
```

- Para API obter detalhes, consulte [DescribeStackSet](#) na Referência de AWS CLI Comandos.

describe-stacks

O código de exemplo a seguir mostra como usar `describe-stacks`.

AWS CLI

Para descrever AWS CloudFormation pilhas

O comando `describe-stacks` a seguir mostra informações resumidas da pilha `myteststack`:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Saída:

```
{
  "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
    }
]
}
```

Para obter mais informações, consulte Stacks no Guia do AWS CloudFormation usuário.

- Para API obter detalhes, consulte [DescribeStacks](#) na Referência de AWS CLI Comandos.

describe-type-registration

O código de exemplo a seguir mostra como usar `describe-type-registration`.

AWS CLI

Para exibir as informações de registro do tipo

O `describe-type-registration` exemplo a seguir exibe informações sobre o registro de tipo especificado, incluindo o status atual, o tipo e a versão do tipo.

```
aws cloudformation describe-type-registration \
  --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Usando o CloudFormation Registro](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [DescribeTypeRegistration](#) na Referência de AWS CLI Comandos.

describe-type

O código de exemplo a seguir mostra como usar `describe-type`.

AWS CLI

Para exibir informações de tipo

O `describe-type` exemplo a seguir exibe informações para o tipo especificado.

```
aws cloudformation describe-type \  
  --type-name My::Logs::LogGroup \  
  --type RESOURCE
```

Saída:

```
{  
  "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]"  
}
```

Para obter mais informações, consulte [Usando o CloudFormation Registro](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [DescribeType](#) na Referência de AWS CLI Comandos.

detect-stack-drift

O código de exemplo a seguir mostra como usar `detect-stack-drift`.

AWS CLI

Para detectar recursos desviados

O `detect-stack-drift` exemplo a seguir inicia a detecção de desvio para a pilha especificada.

```
aws cloudformation detect-stack-drift \  
  --stack-name my-stack
```

Saída:

```
{  
  "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"  
}
```

Em seguida, você pode usar esse ID com o `describe-stack-resource-drifts` comando para descrever recursos desviados.

- Para API obter detalhes, consulte [DetectStackDrift](#) na Referência de AWS CLI Comandos.

`detect-stack-resource-drift`

O código de exemplo a seguir mostra como usar `detect-stack-resource-drift`.

AWS CLI

Para detectar o desvio de um recurso

O `detect-stack-resource-drift` exemplo a seguir verifica um recurso nomeado `MyFunction` em uma pilha chamada `MyStack` de drift:

```
aws cloudformation detect-stack-resource-drift \  
  --stack-name MyStack \  
  --logical-resource-id MyFunction
```

A saída mostra uma função AWS Lambda que foi modificada: `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"
}
}

```

- Para API obter detalhes, consulte [DetectStackResourceDrift](#) na Referência de AWS CLI Comandos.

detect-stack-set-drift

O código de exemplo a seguir mostra como usar `detect-stack-set-drift`.

AWS CLI

Para detectar desvios em um conjunto de pilhas e em todas as instâncias de pilha associadas

O `detect-stack-set-drift` exemplo a seguir inicia as operações de detecção de desvio no conjunto de pilhas especificado, incluindo todas as instâncias de pilha associadas a esse conjunto de pilhas, e retorna um ID de operação que pode ser usado para rastrear o status da operação de desvio.

```
aws cloudformation detect-stack-set-drift \  
  --stack-set-name stack-set-drift-example
```

Saída:

```
{  
  "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE111111"  
}
```

Para obter mais informações, consulte [Detectando alterações de configuração não gerenciadas em conjuntos de pilhas no Guia](#) do AWS CloudFormation usuário.

- Para API obter detalhes, consulte [DetectStackSetDrift](#) na Referência de AWS CLI Comandos.

`estimate-template-cost`

O código de exemplo a seguir mostra como usar `estimate-template-cost`.

AWS CLI

Para estimar o custo do modelo

O exemplo de `estimate-template-cost` a seguir gera uma estimativa de custo para um modelo denominado `template.yaml` na pasta atual.

```
aws cloudformation estimate-template-cost \  
  --template-body file://template.yaml
```

Saída:

```
{
```

```
"Url": "http://calculator.s3.amazonaws.com/calc5.html?
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"
}
```

- Para API obter detalhes, consulte [EstimateTemplateCost](#) na Referência de AWS CLI Comandos.

execute-change-set

O código de exemplo a seguir mostra como usar `execute-change-set`.

AWS CLI

Para executar um conjunto de alterações

O `execute-change-set` exemplo a seguir executa um conjunto de alterações especificado pelo nome do conjunto de alterações e nome da pilha.

```
aws cloudformation execute-change-set \
  --change-set-name my-change-set \
  --stack-name my-stack
```

O `execute-change-set` exemplo a seguir executa um conjunto de alterações especificado pela totalidade ARN do conjunto de alterações.

```
aws cloudformation execute-change-set \
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- Para API obter detalhes, consulte [ExecuteChangeSet](#) na Referência de AWS CLI Comandos.

get-stack-policy

O código de exemplo a seguir mostra como usar `get-stack-policy`.

AWS CLI

Para ver uma política de pilha

O `get-stack-policy` exemplo a seguir exibe a política de pilha para a pilha especificada. Para anexar uma política a uma pilha, use o `set-stack-policy` comando.

```
aws cloudformation get-stack-policy \
  --stack-name my-stack
```

Saída:

```
{
  "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}"
```

- Para API obter detalhes, consulte [GetStackPolicy](#) na Referência de AWS CLI Comandos.

get-template-summary

O código de exemplo a seguir mostra como usar `get-template-summary`.

AWS CLI

Para exibir um resumo do modelo

O comando a seguir exibe informações resumidas sobre os recursos e metadados do arquivo de modelo especificado.

```
aws cloudformation get-template-summary \
  --template-body file://template.yaml
```

Saída:

```
{
  "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"
}

```

- Para API obter detalhes, consulte [GetTemplateSummary](#) na Referência de AWS CLI Comandos.

get-template

O código de exemplo a seguir mostra como usar `get-template`.

AWS CLI

Para visualizar o corpo do modelo de uma AWS CloudFormation pilha

O comando `get-template` a seguir mostra o modelo da pilha `myteststack`:

```
aws cloudformation get-template --stack-name myteststack
```

Saída:

```

{
  "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": {

```


list-exports

O código de exemplo a seguir mostra como usar `list-exports`.

AWS CLI

Para listar as exportações

O `list-exports` exemplo a seguir exibe uma lista das exportações das pilhas na região atual.

```
aws cloudformation list-exports
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListExports](#) na Referência de AWS CLI Comandos.

list-imports

O código de exemplo a seguir mostra como usar `list-imports`.

AWS CLI

Para listar as importações

O `list-imports` exemplo a seguir lista as pilhas que importam a exportação especificada. Para obter a lista de exportações disponíveis, use o `list-exports` comando.

```
aws cloudformation list-imports \  
  --export-name private-vpc-vpcid
```

Saída:

```
{  
  "Imports": [  
    "my-database-stack"  
  ]  
}
```

- Para API obter detalhes, consulte [ListImports](#) na Referência de AWS CLI Comandos.

list-stack-instances

O código de exemplo a seguir mostra como usar `list-stack-instances`.

AWS CLI

Para listar instâncias de uma pilha

O `list-stack-instances` exemplo a seguir lista as instâncias criadas a partir do conjunto de pilhas especificado.

```
aws cloudformation list-stack-instances \  
  --stack-set-name enable-config
```

O exemplo de saída inclui detalhes sobre uma pilha que falhou na atualização devido a um erro:

```
{  
  "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,
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).\"
  }
]
}

```

- Para API obter detalhes, consulte [ListStackInstances](#) na Referência de AWS CLI Comandos.

list-stack-resources

O código de exemplo a seguir mostra como usar `list-stack-resources`.

AWS CLI

Para listar os recursos de uma pilha

O comando a seguir exibe a lista dos recursos da pilha especificada.

```

aws cloudformation list-stack-resources \
  --stack-name my-stack

```

Saída:

```

{

```

```

"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-HIZXMPLE0M9E",
    "ResourceType": "AWS::IAM::Role",
    "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
    "ResourceStatus": "CREATE_COMPLETE",
    "DriftInformation": {
      "StackResourceDriftStatus": "IN_SYNC"
    }
  }
]
}

```

- Para API obter detalhes, consulte [ListStackResources](#) na Referência de AWS CLI Comandos.

list-stack-set-operation-results

O código de exemplo a seguir mostra como usar `list-stack-set-operation-results`.

AWS CLI

Para listar os resultados da operação do conjunto de pilhas

O comando a seguir exibe os resultados de uma operação de atualização em instâncias no conjunto de pilhas especificado.

```
aws cloudformation list-stack-set-operation-results \  
  --stack-set-name enable-config \  
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

Saída:

```
{  
  "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: O SKIPPED status de AccountGateResult é esperado para operações bem-sucedidas, a menos que você crie uma função de bloqueio de conta.

- Para API obter detalhes, consulte [ListStackSetOperationResults](#) na Referência de AWS CLI Comandos.

list-stack-set-operations

O código de exemplo a seguir mostra como usar list-stack-set-operations.

AWS CLI

Para listar as operações do conjunto de pilhas

O `list-stack-set-operations` exemplo a seguir exibe a lista das operações mais recentes no conjunto de pilhas especificado.

```
aws cloudformation list-stack-set-operations \  
  --stack-set-name my-stack-set
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListStackSetOperations](#) na Referência de AWS CLI Comandos.

list-stack-sets

O código de exemplo a seguir mostra como usar `list-stack-sets`.

AWS CLI

Para listar conjuntos de pilhas

O `list-stack-sets` exemplo a seguir exibe a lista de conjuntos de pilhas na região e na conta atuais.

```
aws cloudformation list-stack-sets
```

Saída:

```
{
  "Summaries": [
    {
      "StackSetName": "enable-config",
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
      "Description": "Enable AWS Config",
      "Status": "ACTIVE"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListStackSets](#) na Referência de AWS CLI Comandos.

list-stacks

O código de exemplo a seguir mostra como usar `list-stacks`.

AWS CLI

Para listar AWS CloudFormation pilhas

O comando `list-stacks` a seguir mostra um resumo de todas as pilhas que têm um status de `CREATE_COMPLETE`:

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Saída:

```
[
  {
    "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"
}
]
```

- Para API obter detalhes, consulte [ListStacks](#) na Referência de AWS CLI Comandos.

list-type-registrations

O código de exemplo a seguir mostra como usar `list-type-registrations`.

AWS CLI

Para listar os registros concluídos de um tipo

O `list-type-registrations` exemplo a seguir exibe uma lista dos registros de tipo concluídos para o tipo especificado.

```
aws cloudformation list-type-registrations \
  --type RESOURCE \
  --type-name My::Logs::LogGroup \
  --registration-status-filter COMPLETE
```

Saída:

```
{
  "RegistrationTokenList": [
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
  ]
}
```

Para obter mais informações, consulte [Usando o CloudFormation Registro](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [ListTypeRegistrations](#) na Referência de AWS CLI Comandos.

list-type-versions

O código de exemplo a seguir mostra como usar `list-type-versions`.

AWS CLI

Para listar a versão de uma extensão

O `list-type-versions` exemplo a seguir retorna informações resumidas sobre as versões de uma extensão.

```
aws cloudformation list-type-versions \
  --endpoint https://example.com \
  --region us-west-2 \
  --type RESOURCE \
  --type-name My::Resource::Example \
  --publisher-id 123456789012
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [ListTypeVersions](#) na Referência de AWS CLI Comandos.

list-types

O código de exemplo a seguir mostra como usar `list-types`.

AWS CLI

Para listar os tipos de recursos privados em uma conta

O `list-types` exemplo a seguir exibe uma lista dos tipos de recursos privados atualmente registrados na AWS conta atual.

```
aws cloudformation list-types
```

Saída:

```
{
```

```

    "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Usando o CloudFormation Registro](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [ListTypes](#) na Referência de AWS CLI Comandos.

package

O código de exemplo a seguir mostra como usar package.

AWS CLI

O comando a seguir exporta um modelo chamado `template.json` carregando artefatos locais para o bucket do S3 `bucket-name` e grava o modelo exportado em: `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
```

- Para API obter detalhes, consulte [Package](#) in AWS CLI Command Reference.

publish-type

O código de exemplo a seguir mostra como usar `publish-type`.

AWS CLI

Para publicar uma extensão

O `publish-type` exemplo a seguir publica a extensão especificada no CloudFormation registro como uma extensão pública nessa região.

```
aws cloudformation publish-type \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::1234567890abcdef0
```

Saída:

```
{  
  "PublicTypeArn": "arn:aws:cloudformation:us-west-2::type/  
resource/000q6TfUovXsEMmgKowxDZLLwqr2QUshd2e75c8c/Example-  
Test-1234567890abcdef0/1.0.0"  
}
```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [PublishType](#) na Referência de AWS CLI Comandos.

register-publisher

O código de exemplo a seguir mostra como usar `register-publisher`.

AWS CLI

Para registrar um editor

O `register-publisher` exemplo a seguir registra um editor e aceita o parâmetro de termos e condições.

```
aws cloudformation register-publisher \  

```

```
--region us-west-2 \  
--accept-terms-and-conditions
```

Saída:

```
{  
  "PublisherId": "000q6TfUovXsEMmgKowxDZLlWqr2QUshd2e75c8c"  
}
```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [RegisterPublisher](#) na Referência de AWS CLI Comandos.

register-type

O código de exemplo a seguir mostra como usar `register-type`.

AWS CLI

Para registrar um tipo de recurso

O `register-type` exemplo a seguir registra o tipo de recurso especificado como um tipo de recurso privado na conta do usuário.

```
aws cloudformation register-type \  
  --type-name My::Organization::ResourceName \  
  --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \  
  --type RESOURCE
```

Saída:

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"  
}
```

Para obter mais informações, consulte [Registrando provedores de recursos](#) no Guia do usuário da interface de linha de CloudFormation comando para desenvolvimento de tipos.

- Para API obter detalhes, consulte [RegisterType](#) na Referência de AWS CLI Comandos.

set-stack-policy

O código de exemplo a seguir mostra como usar `set-stack-policy`.

AWS CLI

Para aplicar uma política de pilha

O `set-stack-policy` exemplo a seguir desativa as atualizações para o recurso especificado na pilha especificada. `stack-policy.json` é um JSON documento que define as operações permitidas nos recursos na pilha.

```
aws cloudformation set-stack-policy \  
  --stack-name my-stack \  
  --stack-policy-body file://stack-policy.json
```

Saída:

```
{  
  "Statement" : [  
    {  
      "Effect" : "Allow",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*"   
    },  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "LogicalResourceId/bucket"   
    }  
  ]  
}
```

- Para API obter detalhes, consulte [SetStackPolicy](#) na Referência de AWS CLI Comandos.

set-type-configuration

O código de exemplo a seguir mostra como usar `set-type-configuration`.

AWS CLI

Para configurar dados

O `set-type-configuration` exemplo a seguir especifica os dados de configuração de uma CloudFormation extensão registrada, em determinada conta e região.

```
aws cloudformation set-type-configuration \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::Type \  
  --configuration-alias default \  
  --configuration '{"CredentialKey": "testUserCredential"}'
```

Saída:

```
{  
  "ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-  
configuration/resource/Example-Test-Type/default"  
}
```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [SetTypeConfiguration](#) na Referência de AWS CLI Comandos.

set-type-default-version

O código de exemplo a seguir mostra como usar `set-type-default-version`.

AWS CLI

Para definir a versão padrão de um tipo

O `set-type-default-version` exemplo a seguir define a versão do tipo especificado a ser usada como padrão para esse tipo.

```
aws cloudformation set-type-default-version \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --version-id 00000003
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o CloudFormation Registro](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [SetTypeDefaultVersion](#) na Referência de AWS CLI Comandos.

signal-resource

O código de exemplo a seguir mostra como usar `signal-resource`.

AWS CLI

Para sinalizar um recurso

O `signal-resource` exemplo a seguir indica `success` o cumprimento da condição de espera nomeada `MyWaitCondition` na pilha chamada `my-stack`.

```
aws cloudformation signal-resource \  
  --stack-name my-stack \  
  --logical-resource-id MyWaitCondition \  
  --unique-id 1234 \  
  --status SUCCESS
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SignalResource](#) na Referência de AWS CLI Comandos.

stop-stack-set-operation

O código de exemplo a seguir mostra como usar `stop-stack-set-operation`.

AWS CLI

Para interromper uma operação de conjunto de pilhas

O `stop-stack-set-operation` exemplo a seguir interrompe uma operação de atualização em andamento no conjunto de pilhas especificado.

```
aws cloudformation stop-stack-set-operation \  
  --stack-set-name my-stack-set \  
  --unique-id 1234
```

```
--operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

Este comando não produz saída.

- Para API obter detalhes, consulte [StopStackSetOperation](#) na Referência de AWS CLI Comandos.

test-type

O código de exemplo a seguir mostra como usar `test-type`.

AWS CLI

Para testar uma extensão

O `test-type` exemplo a seguir testa uma extensão registrada para garantir que ela atenda a todos os requisitos necessários para ser publicada no CloudFormation registro.

```
aws cloudformation test-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-  
Resource123/00000001
```

Saída:

```
{  
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/  
Sample-Test-Resource123/00000001"  
}
```

Para obter mais informações, consulte [Usando o AWS CloudFormation registro](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [TestType](#) na Referência de AWS CLI Comandos.

update-stack-instances

O código de exemplo a seguir mostra como usar `update-stack-instances`.

AWS CLI

Para atualizar instâncias de pilha

O `update-stack-instances` exemplo a seguir repete uma atualização nas instâncias de pilha em duas contas em duas regiões com as configurações mais recentes. A configuração de tolerância a falhas especificada garante que a atualização seja tentada em todas as contas e regiões, mesmo que algumas pilhas não possam ser atualizadas.

```
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
```

Saída:

```
{  
  "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"  
}
```

- Para API obter detalhes, consulte [UpdateStackInstances](#) na Referência de AWS CLI Comandos.

update-stack-set

O código de exemplo a seguir mostra como usar `update-stack-set`.

AWS CLI

Para atualizar um conjunto de pilhas

O `update-stack-set` exemplo a seguir adiciona uma tag com o nome da chave `Owner` e um valor de `IT` às instâncias da pilha no conjunto de pilhas especificado.

```
aws cloudformation update-stack-set \  
  --stack-set-name my-stack-set \  
  --use-previous-template \  
  --tags Key=Owner,Value=IT
```

Saída:

```
{  
  "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"
```

```
}
```

- Para API obter detalhes, consulte [UpdateStackSet](#) na Referência de AWS CLI Comandos.

update-stack

O código de exemplo a seguir mostra como usar `update-stack`.

AWS CLI

Para atualizar as AWS CloudFormation pilhas

O comando `update-stack` a seguir atualiza o modelo e os parâmetros de entrada para a pilha `mystack`:

```
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
```

O comando `update-stack` a seguir atualiza apenas o valor do parâmetro `SubnetIDs` da pilha `mystack`. Se você não especificar um valor para o parâmetro, o valor padrão especificado no modelo será usado:

```
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
```

O comando `update-stack` a seguir adiciona à pilha `mystack` dois tópicos de notificação de pilha:

```
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"
```

Para obter mais informações, consulte [atualizações de AWS CloudFormation pilha](#) no Guia do AWS CloudFormation usuário.

- Para API obter detalhes, consulte [UpdateStack](#) na Referência de AWS CLI Comandos.

update-termination-protection

O código de exemplo a seguir mostra como usar `update-termination-protection`.

AWS CLI

Para ativar a proteção contra rescisão

O `update-termination-protection` exemplo a seguir ativa a proteção contra encerramento na pilha especificada.

```
aws cloudformation update-termination-protection \  
  --stack-name my-stack \  
  --enable-termination-protection
```

Saída:

```
{  
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/  
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"  
}
```

- Para API obter detalhes, consulte [UpdateTerminationProtection](#) na Referência de AWS CLI Comandos.

validate-template

O código de exemplo a seguir mostra como usar `validate-template`.

AWS CLI

Para validar um modelo AWS CloudFormation

O comando `validate-template` a seguir valida o modelo `sampletemplate.json`:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Saída:

```
{  
  "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": []
}
```

Para obter mais informações, consulte [Trabalhando com AWS CloudFormation modelos](#) no Guia AWS CloudFormation do usuário.

- Para API obter detalhes, consulte [ValidateTemplate](#) na Referência de AWS CLI Comandos.

CloudFront exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CloudFront.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-cloud-front-origin-access-identity

O código de exemplo a seguir mostra como usar `create-cloud-front-origin-access-identity`.

AWS CLI

Para criar uma identidade de acesso de CloudFront origem

O exemplo a seguir cria uma identidade de acesso de CloudFront origem (OAI) fornecendo a OAI configuração como um argumento de linha de comando:

```
aws cloudfront create-cloud-front-origin-access-identity \
```

```
--cloud-front-origin-access-identity-config \
  CallerReference="cli-example",Comment="Example OAI"
```

Você pode fazer a mesma coisa fornecendo a OAI configuração em um JSON arquivo, conforme mostrado no exemplo a seguir:

```
aws cloudfront create-cloud-front-origin-access-identity \
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

O arquivo `OAI-config.json` é um JSON documento no diretório atual que contém o seguinte:

```
{
  "CallerReference": "cli-example",
  "Comment": "Example OAI"
}
```

Se você fornecer a OAI configuração com um argumento de linha de comando ou um JSON arquivo, a saída será a mesma:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/origin-access-identity/
cloudfront/E74FTE3AEXAMPLE",
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI"
    }
  }
}
```

- Para API obter detalhes, consulte [CreateCloudFrontOriginAccessIdentity](#) na Referência de AWS CLI Comandos.

create-distribution-with-tags

O código de exemplo a seguir mostra como usar `create-distribution-with-tags`.

AWS CLI

Para criar uma CloudFront distribuição com tags

O exemplo a seguir cria uma distribuição com duas tags fornecendo a configuração de distribuição e as tags em um JSON arquivo chamado `dist-config-with-tags.json`:

```
aws cloudfront create-distribution-with-tags \  
--distribution-config-with-tags file://dist-config-with-tags.json
```

O arquivo `dist-config-with-tags.json` é um JSON documento na pasta atual que contém o seguinte. Observe o Tags objeto na parte superior do arquivo, que contém duas tags:

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
}
}
```

Saída:

```
{
  "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
}
}
}

```

- Para API obter detalhes, consulte [CreateDistributionWithTags](#) na Referência de AWS CLI Comandos.

create-distribution

O código de exemplo a seguir mostra como usar create-distribution.

AWS CLI

Para criar uma CloudFront distribuição

O exemplo a seguir cria uma distribuição para um bucket do S3 chamado `awsexamplebucket` e também especifica `index.html` como o objeto raiz padrão, usando argumentos de linha de comando:

```
aws cloudfront create-distribution \  
  --origin-domain-name awsexamplebucket.s3.amazonaws.com \  
  --default-root-object index.html
```

Em vez de usar argumentos de linha de comando, você pode fornecer a configuração de distribuição em um JSON arquivo, conforme mostrado no exemplo a seguir:

```
aws cloudfront create-distribution \  
  --distribution-config file://dist-config.json
```

O arquivo `dist-config.json` é um JSON documento na pasta atual que contém o seguinte:

```
{  
  "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
}

```

Se você fornecer as informações de distribuição com um argumento de linha de comando ou um JSON arquivo, a saída será a mesma:

```

{
  "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
  }
}

```

- Para API obter detalhes, consulte [CreateDistribution](#) na Referência de AWS CLI Comandos.

create-field-level-encryption-config

O código de exemplo a seguir mostra como usar `create-field-level-encryption-config`.

AWS CLI

Para criar uma configuração de CloudFront criptografia em nível de campo

O exemplo a seguir cria uma configuração de criptografia em nível de campo fornecendo os parâmetros de configuração em um JSON arquivo chamado `fle-config.json`. Antes de criar uma configuração de criptografia em nível de campo, você deve ter um perfil de criptografia em nível de campo. Para criar um perfil, consulte o comando `create-field-level-encryption -profile`.

Para obter mais informações sobre criptografia em CloudFront nível de campo, consulte [Usando criptografia em nível de campo para ajudar a proteger dados confidenciais no](#) Amazon Developer Guide. CloudFront

```

aws cloudfront create-field-level-encryption-config \
  --field-level-encryption-config file://fle-config.json

```

O arquivo `fle-config.json` é um JSON documento na pasta atual que contém o seguinte:

```

{
  "CallerReference": "cli-example",
  "Comment": "Example FLE configuration",
  "QueryArgProfileConfig": {

```

```

    "ForwardWhenQueryArgProfileIsUnknown": true,
    "QueryArgProfiles": {
      "Quantity": 0
    }
  },
  "ContentTypeProfileConfig": {
    "ForwardWhenContentTypeIsUnknown": true,
    "ContentTypeProfiles": {
      "Quantity": 1,
      "Items": [
        {
          "Format": "URLEncoded",
          "ProfileId": "P280MFCLSYOCVU",
          "ContentType": "application/x-www-form-urlencoded"
        }
      ]
    }
  }
}

```

Saída:

```

{
  "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,

```



```

"CallerReference": "cli-example",
"Comment": "FLE profile for AWS CLI example",
"EncryptionEntities": {
  "Quantity": 1,
  "Items": [
    {
      "PublicKeyId": "K2K8NC4HVFE3M0",
      "ProviderId": "ExampleFLEProvider",
      "FieldPatterns": {
        "Quantity": 1,
        "Items": [
          "ExampleSensitiveField"
        ]
      }
    }
  ]
}
}
}
}

```

Saída:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption-profile/PPK0UOSIF5WSV",
  "ETag": "E2QWRUHEXAMPLE",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0UOSIF5WSV",
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfile",
      "CallerReference": "cli-example",
      "Comment": "FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {
            "PublicKeyId": "K2K8NC4HVFE3M0",
            "ProviderId": "ExampleFLEProvider",
            "FieldPatterns": {
              "Quantity": 1,
              "Items": [
                "ExampleSensitiveField"
              ]
            }
          }
        ]
      }
    }
  }
}

```



```

    },
    "CallerReference": "cli-1575570291-670203"
  }
}

```

No exemplo anterior, o gerou AWS CLI automaticamente um aleatório `CallerReference`. Para especificar seus próprios `CallerReference` argumentos ou evitar passar os parâmetros de invalidação como argumentos da linha de comando, você pode usar um JSON arquivo. O exemplo a seguir cria uma invalidação para dois arquivos, fornecendo os parâmetros de invalidação em um JSON arquivo chamado: `inv-batch.json`

```

aws cloudfront create-invalidation \
  --distribution-id EDFDVBD6EXAMPLE \
  --invalidation-batch file://inv-batch.json

```

Conteúdo de `inv-batch.json`:

```

{
  "Paths": {
    "Quantity": 2,
    "Items": [
      "/example-path/example-file.jpg",
      "/example-path/example-file2.png"
    ]
  },
  "CallerReference": "cli-example"
}

```

Saída:

```

{
  "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"
}
}
}

```

- Para API obter detalhes, consulte [CreateInvalidation](#) na Referência de AWS CLI Comandos.

create-public-key

O código de exemplo a seguir mostra como usar `create-public-key`.

AWS CLI

Para criar uma chave CloudFront pública

O exemplo a seguir cria uma chave CloudFront pública fornecendo os parâmetros em um JSON arquivo chamado `pub-key-config.json`. Antes de usar esse comando, você deve ter uma chave pública PEM codificada em -. Para obter mais informações, consulte [Criar um par de RSA chaves](#) no Amazon CloudFront Developer Guide.

```

aws cloudfront create-public-key \
  --public-key-config file://pub-key-config.json

```

O arquivo `pub-key-config.json` é um JSON documento na pasta atual que contém o seguinte. Observe que a chave pública é codificada em PEM formato.

```

{
  "CallerReference": "cli-example",
  "Name": "ExampleKey",
  "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAxPmBCA2Ks01nd7IR+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"
}
```

Saída:

```
{
  "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-----
\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"
    }
  }
}
```

- Para API obter detalhes, consulte [CreatePublicKey](#) na Referência de AWS CLI Comandos.

delete-cloud-front-origin-access-identity

O código de exemplo a seguir mostra como usar `delete-cloud-front-origin-access-identity`.

AWS CLI

Para excluir uma identidade de acesso de CloudFront origem

O exemplo a seguir exclui a identidade de acesso de origem (OAI) com o `IDE74FTE3AEXAMPLE`. Para excluir um OAI, você deve ter OAI o ID ETag e. O OAI ID é retornado na saída dos comandos `create-cloud-front-origin -access-identity` e `list-cloud-front-origin -access-identities`.

Para obter o ETag, use o comando `get-cloud-front-origin -access-identity` ou `get-cloud-front-origin -access-identity-config`. Use a `--if-match` opção de fornecer OAI o ETag.

```
aws cloudfront delete-cloud-front-origin-access-identity \  
  --id E74FTE3AEXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

Quando bem-sucedido, este comando não tem saída.

- Para API obter detalhes, consulte [DeleteCloudFrontOriginAccessIdentity](#) na Referência de AWS CLI Comandos.

delete-distribution

O código de exemplo a seguir mostra como usar `delete-distribution`.

AWS CLI

Para excluir uma CloudFront distribuição

O exemplo a seguir exclui a CloudFront distribuição com o `IDEDFDVBD6EXAMPLE`. Você deve desabilitar uma distribuição antes de excluí-la. Use o comando `update-distribution` para desabilitar uma distribuição. Para obter mais informações, consulte exemplos do comando `update-distribution`.

Uma distribuição pode ser excluída depois de ser desabilitada. Para excluir uma distribuição, use a opção `--if-match` para fornecer o ETag da distribuição. Para obter o ETag, use o comando `get-distribution` ou `get-distribution-config`.

```
aws cloudfront delete-distribution \  
  --id EDFDVBD6EXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

Quando bem-sucedido, este comando não tem saída.

- Para API obter detalhes, consulte [DeleteDistribution](#) na Referência de AWS CLI Comandos.

delete-field-level-encryption-config

O código de exemplo a seguir mostra como usar `delete-field-level-encryption-config`.

AWS CLI

Para excluir uma configuração de CloudFront criptografia em nível de campo

O exemplo a seguir exclui a configuração de criptografia CloudFront em nível de campo com o ID. C3KM2WVD605UAY Para excluir uma configuração de criptografia em nível de campo, você deve ter sua ID e. ETag O ID é retornado na saída dos comandos `create-field-level-encryption -config` e `list-field-level-encryption -configs`. Para obter oETag, use o comando `get-field-level-encryption` or `get-field-level-encryption -config`. Use a `--if-match` opção para fornecer as configuraçõesETag.

```
aws cloudfront delete-field-level-encryption-config \  
  --id C3KM2WVD605UAY \  
  --if-match E26M4BIAV81ZF6
```

Quando bem-sucedido, este comando não tem saída.

- Para API obter detalhes, consulte [DeleteFieldLevelEncryptionConfig](#) na Referência de AWS CLI Comandos.

`delete-field-level-encryption-profile`

O código de exemplo a seguir mostra como usar `delete-field-level-encryption-profile`.

AWS CLI

Para excluir um perfil de CloudFront criptografia em nível de campo

O exemplo a seguir exclui o perfil de criptografia CloudFront em nível de campo com o ID. PPK0U0SIF5WSV Para excluir um perfil de criptografia em nível de campo, você deve ter seu ID e. ETag O ID é retornado na saída dos comandos `create-field-level-encryption -profile` e `list-field-level-encryption -profiles`. Para obter oETag, use o comando `get-field-level-encryption -profile` ou `get-field-level-encryption -profile-config`. Use a `--if-match` opção para fornecer o perfilETag.

```
aws cloudfront delete-field-level-encryption-profile \  
  --id PPK0U0SIF5WSV \  
  --if-match EJETYFJ9CL66D
```

Quando bem-sucedido, este comando não tem saída.

- Para API obter detalhes, consulte [DeleteFieldLevelEncryptionProfile](#) na Referência de AWS CLI Comandos.

delete-public-key

O código de exemplo a seguir mostra como usar `delete-public-key`.

AWS CLI

Para excluir uma chave CloudFront pública

O exemplo a seguir exclui a chave CloudFront pública com o `IDKDFB19YGCR002`. Para excluir uma chave pública, você deve ter seu ID ETag e. O ID é retornado na saída dos `list-public-keys` comandos `create-public-key` e. Para obter o ETag, use o `get-public-key-config` comando `get-public-key` or. Use a `--if-match` opção de fornecer a chave pública ETag.

```
aws cloudfront delete-public-key \  
  --id KDFB19YGCR002 \  
  --if-match E2QWRUHEXAMPLE
```

Quando bem-sucedido, este comando não tem saída.

- Para API obter detalhes, consulte [DeletePublicKey](#) na Referência de AWS CLI Comandos.

get-cloud-front-origin-access-identity-config

O código de exemplo a seguir mostra como usar `get-cloud-front-origin-access-identity-config`.

AWS CLI

Para obter uma configuração de identidade de acesso de CloudFront origem

O exemplo a seguir obtém metadados sobre a identidade de acesso de CloudFront origem (OAI) com o `IDE74FTE3AEXAMPLE`, incluindo seu ETag. O OAI ID é retornado na saída dos comandos `create-cloud-front-origin -access-identity` e `list-cloud-front-origin -access-identities`.

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

Saída:

```
{  
  "ETag": "E2QWRUHEXAMPLE",
```

```
"CloudFrontOriginAccessIdentityConfig": {
  "CallerReference": "cli-example",
  "Comment": "Example OAI"
}
}
```

- Para API obter detalhes, consulte [GetCloudFrontOriginAccessIdentityConfig](#) na Referência de AWS CLI Comandos.

get-cloud-front-origin-access-identity

O código de exemplo a seguir mostra como usar `get-cloud-front-origin-access-identity`.

AWS CLI

Para obter uma identidade de acesso de CloudFront origem

O exemplo a seguir obtém a identidade de acesso de CloudFront origem (OAI) com a `E74FTE3AEXAMPLE`, incluindo sua ETag e a ID canônica associada do S3. O OAI ID é retornado na saída dos comandos `create-cloud-front-origin -access-identity` e `list-cloud-front-origin -access-identities`.

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

Saída:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI"
    }
  }
}
```

- Para API obter detalhes, consulte [GetCloudFrontOriginAccessIdentity](#) na Referência de AWS CLI Comandos.

get-distribution-config

O código de exemplo a seguir mostra como usar `get-distribution-config`.

AWS CLI

Para obter uma configuração CloudFront de distribuição

O exemplo a seguir obtém metadados sobre a CloudFront distribuição com o `IDEDFDVBD6EXAMPLE`, incluindo seu ETag. O ID de distribuição é retornado nos comandos `create-distribution` e `list-distributions`.

```
aws cloudfront get-distribution-config --id EDFDVBD6EXAMPLE
```

Saída:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-cli-example",
          "DomainName": "awsexamplebucket.s3.amazonaws.com",
          "OriginPath": "",
          "CustomHeaders": {
            "Quantity": 0
          },
          "S3OriginConfig": {
            "OriginAccessIdentity": ""
          }
        }
      ]
    },
    "OriginGroups": {
      "Quantity": 0
    },
  },
}
```

```
"DefaultCacheBehavior": {
  "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-example",
  "ForwardedValues": {
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 0
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
```

```

    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
  "Enabled": true,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLId": "",
  "HttpVersion": "http2",
  "IsIPV6Enabled": true
}
}

```

- Para API obter detalhes, consulte [GetDistributionConfig](#) na Referência de AWS CLI Comandos.

get-distribution

O código de exemplo a seguir mostra como usar `get-distribution`.

AWS CLI

Para obter uma CloudFront distribuição

O exemplo a seguir obtém a CloudFront distribuição com o `IDEDFDVBD6EXAMPLE`, incluindo seu `ETag`. O ID de distribuição é retornado nos comandos `create-distribution` e `list-distributions`.

```
aws cloudfront get-distribution --id EDFDVBD6EXAMPLE
```

Saída:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "Deployed",
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",
    "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
  }
}
}
}

```

- Para API obter detalhes, consulte [GetDistribution](#) na Referência de AWS CLI Comandos.

get-field-level-encryption-config

O código de exemplo a seguir mostra como usar `get-field-level-encryption-config`.

AWS CLI

Para obter metadados sobre uma configuração de criptografia em CloudFront nível de campo

O exemplo a seguir obtém metadados sobre a configuração de criptografia CloudFront em nível de campo com o IDC3KM2WVD605UAY, incluindo seu: ETag


```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

Saída:

```
{
  "ETag": "E2P4Z4VU7TY5SG",
  "FieldLevelEncryptionConfig": {
    "CallerReference": "cli-example",
    "Comment": "Example FLE configuration",
    "QueryArgProfileConfig": {
      "ForwardWhenQueryArgProfileIsUnknown": true,
      "QueryArgProfiles": {
        "Quantity": 0,
        "Items": []
      }
    },
    "ContentTypeProfileConfig": {
      "ForwardWhenContentTypeIsUnknown": true,
      "ContentTypeProfiles": {
        "Quantity": 1,
        "Items": [
          {
            "Format": "URLEncoded",
            "ProfileId": "P280MFCLSY0CVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}
```

- Para API obter detalhes, consulte [GetFieldLevelEncryptionConfig](#) na Referência de AWS CLI Comandos.

get-field-level-encryption-profile-config

O código de exemplo a seguir mostra como usar `get-field-level-encryption-profile-config`.

AWS CLI

Para obter uma configuração de CloudFront perfil de criptografia em nível de campo

O exemplo a seguir obtém metadados sobre o perfil de criptografia CloudFront em nível de campo com IDPPK0U0SIF5WSV, incluindo seu: ETag

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0U0SIF5WSV
```

Saída:

```
{
  "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"
            ]
          }
        }
      ]
    }
  }
}
```

- Para API obter detalhes, consulte [GetFieldLevelEncryptionProfileConfig](#) na Referência de AWS CLI Comandos.

get-field-level-encryption-profile

O código de exemplo a seguir mostra como usar `get-field-level-encryption-profile`.

AWS CLI

Para obter um perfil de CloudFront criptografia em nível de campo

O exemplo a seguir obtém o perfil de criptografia CloudFront em nível de campo com IDPPK0U0SIF5WSV, incluindo seu: ETag

```
aws cloudfront get-field-level-encryption-profile --id PPK0U0SIF5WSV
```

Saída:

```
{
  "ETag": "E1QQG65FS2L2GC",
  "FieldLevelEncryptionProfile": {
    "Id": "PPK0U0SIF5WSV",
    "LastModifiedTime": "2019-12-10T01:03:16.537Z",
    "FieldLevelEncryptionProfileConfig": {
      "Name": "ExampleFLEProfile",
      "CallerReference": "cli-example",
      "Comment": "FLE profile for AWS CLI example",
      "EncryptionEntities": {
        "Quantity": 1,
        "Items": [
          {
            "PublicKeyId": "K2K8NC4HVFE3M0",
            "ProviderId": "ExampleFLEProvider",
            "FieldPatterns": {
              "Quantity": 1,
              "Items": [
                "ExampleSensitiveField"
              ]
            }
          }
        ]
      }
    }
  }
}
```

- Para API obter detalhes, consulte [GetFieldLevelEncryptionProfile](#) na Referência de AWS CLI Comandos.

get-field-level-encryption

O código de exemplo a seguir mostra como usar `get-field-level-encryption`.

AWS CLI

Para obter uma configuração de CloudFront criptografia em nível de campo

O exemplo a seguir obtém a configuração de criptografia CloudFront em nível de campo com o `IDC3KM2WVD605UAY`, incluindo seu: `ETag`

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

Saída:

```
{
  "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"
          }
        ]
      }
    }
  }
}
```

```

    }
  }
}

```

- Para API obter detalhes, consulte [GetFieldLevelEncryption](#) na Referência de AWS CLI Comandos.

get-invalidation

O código de exemplo a seguir mostra como usar `get-invalidation`.

AWS CLI

Para obter uma CloudFront invalidação

O exemplo a seguir obtém a invalidação com o ID `I2J0I21PCUY0IK` para a CloudFront distribuição com o ID: `EDFDVBD6EXAMPLE`

```
aws cloudfront get-invalidation --id I2J0I21PCUY0IK --distribution-id EDFDVBD6EXAMPLE
```

Saída:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [GetInvalidation](#) na Referência de AWS CLI Comandos.

get-public-key-config

O código de exemplo a seguir mostra como usar `get-public-key-config`.

AWS CLI

Para obter uma configuração de chave CloudFront pública

O exemplo a seguir obtém metadados sobre a chave CloudFront pública com o IDKDFB19YGCR002, incluindo seuETag. O ID da chave pública é retornado nos `list-public-keys` comandos `create-public-key` e.

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

Saída:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKeyConfig": {
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMwxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
    "Comment": "example public key"
  }
}
```

- Para API obter detalhes, consulte [GetPublicKeyConfig](#) na Referência de AWS CLI Comandos.

get-public-key

O código de exemplo a seguir mostra como usar `get-public-key`.

AWS CLI

Para obter uma chave CloudFront pública

O exemplo a seguir obtém a chave CloudFront pública com o IDKDFB19YGCR002, incluindo seuETag. O ID da chave pública é retornado nos list-public-keys comandos create-public-key e.

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

Saída:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKey": {
    "Id": "KDFB19YGCR002",
    "CreatedTime": "2019-12-05T18:51:43.781Z",
    "PublicKeyConfig": {
      "CallerReference": "cli-example",
      "Name": "ExampleKey",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+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"
    }
  }
}
```

- Para API obter detalhes, consulte [GetPublicKey](#) na Referência de AWS CLI Comandos.

list-cloud-front-origin-access-identities

O código de exemplo a seguir mostra como usar list-cloud-front-origin-access-identities.

AWS CLI

Para listar identidades de acesso de CloudFront origem

O exemplo a seguir obtém uma lista das identidades de acesso de CloudFront origem (OAI) em sua AWS conta:

aws cloudfront list-cloud-front-origin-access-identities

Saída:

```
{
  "CloudFrontOriginAccessIdentityList": {
    "Items": [
      {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "Comment": "Example OAI"
      },
      {
        "Id": "EH1HDMBEXAMPLE",
        "S3CanonicalUserId":
"1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
        "Comment": "Test OAI"
      },
      {
        "Id": "E2X2C9TEXAMPLE",
        "S3CanonicalUserId":
"cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
        "Comment": "Example OAI #2"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [ListCloudFrontOriginAccessIdentities](#) na Referência de AWS CLI Comandos.

list-distributions

O código de exemplo a seguir mostra como usar `list-distributions`.

AWS CLI

Para listar CloudFront distribuições

O exemplo a seguir obtém uma lista das CloudFront distribuições em sua AWS conta:

aws cloudfront list-distributions

Saída:

```
{
  "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": {
        "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
        },
    },
```



```
    },
    "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
    }
  ]
}

```

- Para API obter detalhes, consulte [ListDistributions](#) na Referência de AWS CLI Comandos.

list-field-level-encryption-configs

O código de exemplo a seguir mostra como usar `list-field-level-encryption-configs`.

AWS CLI

Para listar as configurações de CloudFront criptografia em nível de campo

O exemplo a seguir obtém uma lista das configurações de criptografia em CloudFront nível de campo em sua conta: AWS

```
aws cloudfront list-field-level-encryption-configs
```

Saída:


```

{
  "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"
              }
            ]
          }
        }
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ListFieldLevelEncryptionConfigs](#) na Referência de AWS CLI Comandos.

list-field-level-encryption-profiles

O código de exemplo a seguir mostra como usar `list-field-level-encryption-profiles`.

AWS CLI

Para listar perfis de CloudFront criptografia em nível de campo

O exemplo a seguir obtém uma lista dos perfis de criptografia em CloudFront nível de campo em sua AWS conta:

```
aws cloudfront list-field-level-encryption-profiles
```

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListFieldLevelEncryptionProfiles](#) na Referência de AWS CLI Comandos.

list-invalidations

O código de exemplo a seguir mostra como usar `list-invalidations`.

AWS CLI

Para listar as CloudFront invalidações

O exemplo a seguir obtém uma lista das invalidações da CloudFront distribuição com o ID: EDFDVBD6EXAMPLE

```
aws cloudfront list-invalidations --distribution-id EDFDVBD6EXAMPLE
```

Saída:

```

{
  "InvalidationList": {
    "Marker": "",
    "Items": [
      {

```

```

        "Status": "Completed",
        "Id": "YNY2LI2BVJ4NJU",
        "CreateTime": "2019-08-31T21:15:52.042Z"
    }
],
"IsTruncated": false,
"MaxItems": 100,
"Quantity": 1
}
}

```

- Para API obter detalhes, consulte [ListInvalidations](#) na Referência de AWS CLI Comandos.

list-public-keys

O código de exemplo a seguir mostra como usar `list-public-keys`.

AWS CLI

Para listar chaves CloudFront públicas

O exemplo a seguir obtém uma lista das chaves CloudFront públicas em sua AWS conta:

```
aws cloudfront list-public-keys
```

Saída:

```

{
  "PublicKeyList": {
    "MaxItems": 100,
    "Quantity": 2,
    "Items": [
      {
        "Id": "K2K8NC4HVFE3M0",
        "Name": "ExampleKey",
        "CreatedTime": "2019-12-05T01:04:28.818Z",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAxPmbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjm3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq

```

```
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnStb9sr7MIhS6A\nrwIDAQAB\n-----END
PUBLIC KEY-----\n",
      "Comment": "example public key"
    },
    {
      "Id": "K1S0LWQ2L5HTBU",
      "Name": "ExampleKey2",
      "CreatedTime": "2019-12-09T23:28:11.110Z",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCgKCAQEAp0CAg88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1go17FV6pYN178iJg3E08JBbwt1H
+cR9\nLGSf60NDeVhm760c39Np/vWg0dsGQcRbi9WmKZeS0DqjQGzVZWqPmito3FzWV6b
\nfVY5N36U/RdbVAJm95Km+qaMY1bIdF40t72bi3IkKYV5h1B2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmmgIVnk0QM37NYeoDnka0uTpu\nha/
+3b8t0b2z3LBVHPkp85zJRA0XacSwf5rZtPYKBNFsixTa2n55k2r218m0kMC4\nUwIDAQAB\n-----END
PUBLIC KEY-----",
      "Comment": "example public key #2"
    }
  ]
}
}
```

- Para API obter detalhes, consulte [ListPublicKeys](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para uma CloudFront distribuição

O exemplo a seguir obtém uma lista das tags de uma CloudFront distribuição:

```
aws cloudfront list-tags-for-resource \
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE
```

Saída:

```
{
  "Tags": {
    "Items": [
```

```

    {
      "Key": "DateCreated",
      "Value": "2019-12-04"
    },
    {
      "Key": "Name",
      "Value": "Example name"
    },
    {
      "Key": "Project",
      "Value": "Example project"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

sign

O código de exemplo a seguir mostra como usar `sign`.

AWS CLI

Para assinar um CloudFront URL

O exemplo a seguir assina um CloudFront URL. Para assinar um URL, você precisa do ID do par de chaves (chamado de ID da chave de acesso no AWS Management Console) e da chave privada do par de CloudFront chaves do assinante confiável. Para obter mais informações sobre assinatura URLs, consulte Como [servir conteúdo privado com cookies assinados URLs e assinados](#) no 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

```

Saída:

```

https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKvc6pwkif6oZc-

```

```
JWSpDkH0UH7EBGGqvgurkecCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoenH0GFteRf9ELK-
Bs3n1jTLjtbgzIUt7QJNKXcWt8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GG1612kgCpXUB6Jx6K~Y3kpV0dzUP0IqFLHANJojbhxrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FK0qNEaWDIB
GNvjRJxqWf93uMobeM0iVYahb-e0KIItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE
```

- Para API obter detalhes, consulte Referência do AWS CLI comando de [login](#).

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar uma CloudFront distribuição

O `tag-resource` exemplo a seguir adiciona duas tags à CloudFront distribuição especificada.

```
aws cloudfront tag-resource \
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
  --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example project"}]'
```

Em vez de usar argumentos de linha de comando, você pode fornecer as tags em um JSON arquivo, conforme mostrado no exemplo a seguir:

```
aws cloudfront tag-resource \
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
  --tags file://tags.json
```

Conteúdo de `tags.json`:

```
{
  "Items": [
    {
      "Key": "Name",
      "Value": "Example name"
    },
    {
      "Key": "Project",
      "Value": "Example project"
    }
  ]
}
```

```
    }  
  ]  
}
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de uma CloudFront distribuição

O exemplo a seguir remove duas tags de uma CloudFront distribuição usando argumentos de linha de comando:

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys Items=Name,Project
```

Em vez de usar argumentos de linha de comando, você pode fornecer as chaves de tag em um JSON arquivo, conforme mostrado no exemplo a seguir:

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys file://tag-keys.json
```

O arquivo `tag-keys.json` é um JSON documento na pasta atual que contém o seguinte:

```
{  
  "Items": [  
    "Name",  
    "Project"  
  ]  
}
```

Quando bem-sucedido, este comando não tem saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-cloud-front-origin-access-identity

O código de exemplo a seguir mostra como usar `update-cloud-front-origin-access-identity`.

AWS CLI

Para atualizar uma identidade de acesso de CloudFront origem

O exemplo a seguir atualiza a identidade de acesso de origem (OAI) com o `E74FTE3AEXAMPLE`. O único campo que você pode atualizar OAI é o `Comment`.

Para atualizar um OAI, você deve ter OAI o ID ETag e. O OAI ID é retornado na saída dos comandos `create-cloud-front-origin-access-identity` e `list-cloud-front-origin-access-identities`. Para obter o ETag, use o comando `get-cloud-front-origin-access-identity` ou `get-cloud-front-origin-access-identity-config` Use a `--if-match` opção de fornecer OAI o 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"
```

Você pode fazer a mesma coisa fornecendo a OAI configuração em um JSON arquivo, conforme mostrado no exemplo a seguir:

```
aws cloudfront update-cloud-front-origin-access-identity \  
  --id E74FTE3AEXAMPLE \  
  --if-match E2QWRUHEXAMPLE \  
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

O arquivo `OAI-config.json` é um JSON documento no diretório atual que contém o seguinte:

```
{  
  "CallerReference": "cli-example",  
  "Comment": "Example OAI Updated"  
}
```

Se você fornecer a OAI configuração com um argumento de linha de comando ou um JSON arquivo, a saída será a mesma:

```
{
  "ETag": "E9LHASXEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
"cd13868f797c227f7bea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI Updated"
    }
  }
}
```

- Para API obter detalhes, consulte [UpdateCloudFrontOriginAccessIdentity](#) na Referência de AWS CLI Comandos.

update-distribution

O código de exemplo a seguir mostra como usar `update-distribution`.

AWS CLI

Para atualizar o objeto raiz padrão de uma CloudFront distribuição

O exemplo a seguir atualiza o objeto raiz padrão `index.html` para a CloudFront distribuição com o `IDEDFDVBD6EXAMPLE`:

```
aws cloudfront update-distribution --id EDFDVBD6EXAMPLE \
  --default-root-object index.html
```

Saída:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "Distribution": {
    "Id": "EDFDVBD6EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:55:39.870Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
```

```
    "Enabled": false,
    "Quantity": 0
  },
  "DistributionConfig": {
    "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "example-website",
          "DomainName": "www.example.com",
          "OriginPath": "",
          "CustomHeaders": {
            "Quantity": 0
          },
          "CustomOriginConfig": {
            "HTTPPort": 80,
            "HTTPSPort": 443,
            "OriginProtocolPolicy": "match-viewer",
            "OriginSslProtocols": {
              "Quantity": 2,
              "Items": [
                "SSLv3",
                "TLSv1"
              ]
            }
          },
          "OriginReadTimeout": 30,
          "OriginKeepaliveTimeout": 5
        }
      ]
    },
    "OriginGroups": {
      "Quantity": 0
    },
    "DefaultCacheBehavior": {
      "TargetOriginId": "example-website",
      "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
```

```
        "Forward": "none"
    },
    "Headers": {
        "Quantity": 1,
        "Items": [
            "*"
        ]
    },
    "QueryStringCacheKeys": {
        "Quantity": 0
    }
},
"TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
},
"ViewerProtocolPolicy": "allow-all",
"MinTTL": 0,
"AllowedMethods": {
    "Quantity": 2,
    "Items": [
        "HEAD",
        "GET"
    ],
    "CachedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ]
    }
},
"SmoothStreaming": false,
"DefaultTTL": 86400,
"MaxTTL": 31536000,
"Compress": false,
"LambdaFunctionAssociations": {
    "Quantity": 0
},
"FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
},
```

```

    "CustomErrorResponses": {
      "Quantity": 0
    },
    "Comment": "",
    "Logging": {
      "Enabled": false,
      "IncludeCookies": false,
      "Bucket": "",
      "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http1.1",
    "IsIPV6Enabled": true
  }
}

```

Para atualizar uma CloudFront distribuição

O exemplo a seguir desativa a CloudFront distribuição com o ID EMLARXS9EXAMPLE fornecendo a configuração de distribuição em um JSON arquivo chamado `dist-config-disable.json`. Para atualizar uma distribuição, use a opção `--if-match` para fornecer o ETag da distribuição. Para obter o ETag, use o comando `get-distribution` ou `get-distribution-config`.

Depois de usar o exemplo a seguir para desabilitar uma distribuição, use o comando `delete-distribution` para excluí-la.

```

aws cloudfront update-distribution \
  --id EMLARXS9EXAMPLE \
  --if-match E2QWRUHEXAMPLE \

```

```
--distribution-config file://dist-config-disable.json
```

O arquivo `dist-config-disable.json` é um JSON documento na pasta atual que contém o seguinte. Observe que o campo `Enabled` está definido como `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
}
```

Saída:

```
{
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:32:35.553Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
  "DistributionConfig": {
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
```



```
        "DomainName": "awsexamplebucket.s3.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    ]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId":
"awsexamplebucket.s3.amazonaws.com-1574382155-273939",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
        "Headers": {
            "Quantity": 0
        },
        "QueryStringCacheKeys": {
            "Quantity": 0
        }
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
```

```
        "HEAD",
        "GET"
    ]
    }
},
"SmoothStreaming": false,
"DefaultTTL": 86400,
"MaxTTL": 31536000,
"Compress": false,
"LambdaFunctionAssociations": {
    "Quantity": 0
},
"FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": false,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
},
"Restrictions": {
    "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
    }
},
"WebACLId": "",
"HttpVersion": "http2",
"IsIPV6Enabled": true
}
```

```
}
}
```

- Para API obter detalhes, consulte [UpdateDistribution](#) na Referência de AWS CLI Comandos.

update-field-level-encryption-config

O código de exemplo a seguir mostra como usar `update-field-level-encryption-config`.

AWS CLI

Para atualizar uma configuração de CloudFront criptografia em nível de campo

O exemplo a seguir atualiza o `Comment` campo da configuração de criptografia em nível de campo com o ID `C3KM2WVD605UAY` fornecendo os parâmetros em um JSON arquivo.

Para atualizar uma configuração de criptografia em nível de campo, você deve ter o ID da configuração e. ETag O ID é retornado na saída dos comandos `create-field-level-encryption-config` e `list-field-level-encryption-configs`. Para obter o ETag, use o comando `get-field-level-encryption` or `get-field-level-encryption-config`. Use a `--if-match` opção para fornecer as configurações ETag.

```
aws cloudfront update-field-level-encryption-config \
  --id C3KM2WVD605UAY \
  --if-match E2P4Z4VU7TY5SG \
  --field-level-encryption-config file://fle-config.json
```

O arquivo `fle-config.json` é um JSON documento no diretório atual que contém o seguinte:

```
{
  "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"
      }
    ]
  }
}

```

Saída:

```

{
  "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"
          }
        ]
      }
    }
  }
}

```

```
}
}
```

- Para API obter detalhes, consulte [UpdateFieldLevelEncryptionConfig](#) Referência de AWS CLI Comandos.

update-field-level-encryption-profile

O código de exemplo a seguir mostra como usar `update-field-level-encryption-profile`.

AWS CLI

Para atualizar um perfil de CloudFront criptografia em nível de campo

O exemplo a seguir atualiza o perfil de criptografia em nível de campo com o ID.

PPK0U0SIF5WSV Este exemplo atualiza o Name perfil e adiciona um segundo FieldPatterns item, fornecendo os parâmetros em um JSON arquivo. Comment

Para atualizar um perfil de criptografia em nível de campo, você deve ter o ID do perfil e. ETag O ID é retornado na saída dos comandos `create-field-level-encryption -profile` e `list-field-level-encryption -profiles`. Para obter oETag, use o comando `get-field-level-encryption -profile` ou `get-field-level-encryption -profile-config`. Use a `--if-match` opção para fornecer o perfilETag.

```
aws cloudfront update-field-level-encryption-profile \
  --id PPK0U0SIF5WSV \
  --if-match E1QQG65FS2L2GC \
  --field-level-encryption-profile-config file://fle-profile-config.json
```

O arquivo `fle-profile-config.json` é um JSON documento no diretório atual que contém o seguinte:

```
{
  "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"
            ]
        }
    ]
}

```

Saída:

```

{
  "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"
              ]
            }
          }
        ]
      }
    }
  }
}

```

- Para API obter detalhes, consulte [UpdateFieldLevelEncryptionProfile](#) na Referência de AWS CLI Comandos.

CloudSearch Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon CloudSearch.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

upload-documents

O código de exemplo a seguir mostra como usar upload-documents.

AWS CLI

O upload-documents comando a seguir carrega um lote de JSON documentos em um CloudSearch domínio da 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
```

Saída:

```
{
  "status": "success",
  "adds": 5000,
  "deletes": 0
}
```

```
}
```

- Para API obter detalhes, consulte [UploadDocuments](#) na Referência de AWS CLI Comandos.

CloudTrail exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CloudTrail.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags

O código de exemplo a seguir mostra como usar `add-tags`.

AWS CLI

Para adicionar tags à trilha

O `add-tags` comando a seguir adiciona tags para `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
```

- Para API obter detalhes, consulte [AddTags](#) na Referência de AWS CLI Comandos.

create-subscription

O código de exemplo a seguir mostra como usar `create-subscription`.

AWS CLI

Para criar e configurar AWS recursos para uma trilha

O `create-subscription` comando a seguir cria um novo bucket do S3 e um novo SNS tópico para `Trail1`:

```
aws cloudtrail create-subscription --name Trail1 --s3-new-bucket my-bucket --sns-new-topic my-topic
```

Saída:

```
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
```

- Para API obter detalhes, consulte [CreateSubscription](#) na Referência de AWS CLI Comandos.

create-trail

O código de exemplo a seguir mostra como usar `create-trail`.

AWS CLI

Para criar uma trilha

O `create-trail` comando a seguir cria uma trilha multirregional chamada `Trail1` e especifica um bucket do S3:

```
aws cloudtrail create-trail --name Trail1 --s3-bucket-name my-bucket --is-multi-region-trail
```

Saída:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- Para API obter detalhes, consulte [CreateTrail](#) na Referência de AWS CLI Comandos.

delete-trail

O código de exemplo a seguir mostra como usar `delete-trail`.

AWS CLI

Para excluir uma trilha

O `delete-trail` comando a seguir exclui uma trilha chamada `Trail1`:

```
aws cloudtrail delete-trail --name Trail1
```

- Para API obter detalhes, consulte [DeleteTrail](#) na Referência de AWS CLI Comandos.

describe-trails

O código de exemplo a seguir mostra como usar `describe-trails`.

AWS CLI

Para descrever uma trilha

O `describe-trails` comando a seguir retorna as configurações de `Trail1` e `Trail2`:

```
aws cloudtrail describe-trails --trail-name-list Trail1 Trail2
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeTrails](#) na Referência de AWS CLI Comandos.

get-event-selectors

O código de exemplo a seguir mostra como usar `get-event-selectors`.

AWS CLI

Para ver as configurações do seletor de eventos para uma trilha

O `get-event-selectors` comando a seguir retorna as configurações para `Trail1`:

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

Saída:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [],
      "ReadWriteType": "All"
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
}
```

- Para API obter detalhes, consulte [GetEventSelectors](#) na Referência de AWS CLI Comandos.

get-trail-status

O código de exemplo a seguir mostra como usar `get-trail-status`.

AWS CLI

Para obter o status de uma trilha

O `get-trail-status` comando a seguir retorna os detalhes de entrega e registro de `Trail1`:

```
aws cloudtrail get-trail-status --name Trail1
```

Saída:

```
{
```

```

"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"
}

```

- Para API obter detalhes, consulte [GetTrailStatus](#) na Referência de AWS CLI Comandos.

list-public-keys

O código de exemplo a seguir mostra como usar `list-public-keys`.

AWS CLI

Para listar todas as chaves públicas de uma trilha

O `list-public-keys` comando a seguir retorna todas as chaves públicas cujas chaves privadas foram usadas para assinar os arquivos de resumo dentro do intervalo de tempo especificado:

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

Saída:

```

{
  "PublicKeyList": [
    {
      "ValidityStartTime": 1453076702.0,
      "ValidityEndTime": 1455668702.0,
      "Value": "MIIBCgKCAQEAlSS3c192HDycr/MTj0mo0has8habjrxaXw+Kz1WF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kU11xboEcEPHYoIYMbgfSw7KGnuDKwkLzsQWhUJ0cIb0HASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmU18/P7XpfpWQuk4h8g3pXZ0rNXr081bh4d39svj7Uqdhv0XoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",

```

```

    "Fingerprint": "7f3f401420072e50a65a141430817ab3"
  }
]
}

```

- Para API obter detalhes, consulte [ListPublicKeys](#) na Referência de AWS CLI Comandos.

list-tags

O código de exemplo a seguir mostra como usar `list-tags`.

AWS CLI

Para listar as tags de uma trilha

O `list-tags` comando a seguir lista as tags para `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

```

Saída:

```

{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListTags](#) na Referência de AWS CLI Comandos.

lookup-events

O código de exemplo a seguir mostra como usar lookup-events.

AWS CLI

Para pesquisar eventos para uma trilha

O lookup-events comando a seguir pesquisa eventos de API atividade pelo atributoEventName:

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

Saída:

```

{
  "Events": [
    {
      "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
      "Username": "my-session-name",
      "EventTime": "2021-11-18T09:41:02-08:00",
      "CloudTrailEvent": "{\n\"eventVersion\": \"1.02\", \"userIdentity\": {\n\"type\": \"AssumedRole\", \"principalId\": \"AR0AJIKPFTA72SWU4L7T4:my-session-name\", \"arn\": \"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\", \"accountId\": \"123456789012\", \"sessionContext\": {\n\"attributes\": {\n\"mfaAuthenticated\": \"false\", \"creationDate\": \"2016-01-26T21:42:12Z\"}, \"sessionIssuer\": {\n\"type\": \"Role\", \"principalId\": \"AR0AJIKPFTA72SWU4L7T4\", \"arn\": \"arn:aws:iam::123456789012:role/my-role\", \"accountId\": \"123456789012\", \"userName\": \"my-role\"}}}, \"eventTime\": \"2016-01-26T21:42:12Z\", \"eventSource\": \"signin.amazonaws.com\", \"eventName\": \"ConsoleLogin\", \"awsRegion\": \"us-east-1\", \"sourceIPAddress\": \"72.21.198.70\", \"userAgent\": \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36

```

```
(KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36\", \"requestParameters
\":null, \"responseElements\": {\"ConsoleLogin\": {\"Success\": true}, \"additionalEventData\":
 {\"MobileVersion\": \"No\", \"MFAUsed\": \"No\"}, \"eventID\": \"654ccbc0-ba0d-486a-9076-
dbf7274677a7\", \"eventType\": \"AwsConsoleSignIn\", \"recipientAccountId\":
 \"123456789012\"} },
  {
    \"EventName\": \"ConsoleLogin\",
    \"Resources\": []
  }
]
}
```

- Para API obter detalhes, consulte [LookupEvents](#) na Referência de AWS CLI Comandos.

put-event-selectors

O código de exemplo a seguir mostra como usar `put-event-selectors`.

AWS CLI

Exemplo 1: Configurar uma trilha para registrar eventos de gerenciamento e eventos de dados usando seletores de eventos avançados

Você pode adicionar seletores de eventos avançados e condições para seus seletores de eventos avançados, até um máximo de 500 valores para todas as condições e seletores em uma trilha. Você pode usar seletores de eventos avançados para registrar todos os tipos de eventos de dados disponíveis. Você pode usar seletores de eventos avançados ou seletores de eventos básicos, mas não ambos. Se você aplicar seletores de eventos avançados a uma trilha, todos os seletores de eventos básicos existentes serão substituídos.

O exemplo a seguir cria um seletor de eventos avançado para uma trilha chamada `myTrail` para registrar todos os eventos de gerenciamento, registrar o S3 `PutObject` e as `DeleteObject` API chamadas para todos, exceto um bucket do S3, registrar as API chamadas de dados para uma função Lambda chamada `myFunction` e registrar as API chamadas de publicação em um tópico chamado. 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"] }]]}'
```

Saída:

```
{
  "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"
        ]
    }
]
}
]
}

```

Para obter mais informações, consulte [Registrar eventos usando seletores de eventos avançados](#) no Guia do AWS CloudTrail usuário.

Exemplo 2: Configurar seletores de eventos para uma trilha para registrar todos os eventos de gerenciamento e eventos de dados

Você pode configurar até 5 seletores de eventos para uma trilha e até 250 recursos de dados para uma trilha. Os seletores de eventos também são chamados de seletores de eventos básicos. Você pode usar seletores de eventos para registrar eventos de gerenciamento e eventos de dados para objetos do S3, funções Lambda e tabelas do DynamoDB. Para registrar eventos de dados para outros tipos de recursos, você deve usar seletores de eventos avançados.

O exemplo a seguir cria um seletor de eventos para uma trilha nomeada `TrailName` para incluir todos os eventos de gerenciamento, eventos de dados para duas combinações de bucket/prefixo

do Amazon S3 e eventos de dados para uma única função Lambda chamada. 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"]}]]'
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Registrar eventos usando seletores de eventos básicos](#) no Guia do AWS CloudTrail usuário.

Exemplo 3: configure seletores de eventos para uma trilha para registrar eventos de gerenciamento, todos os eventos de dados do S3 em objetos do S3 e todos os eventos de dados do Lambda em funções em sua conta

O exemplo a seguir cria um seletor de eventos para uma trilha chamada `TrailName2` que inclui todos os eventos de gerenciamento e todos os eventos de dados para todos os buckets do Amazon S3 AWS e funções Lambda na conta. 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"]}]]'
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Registrar eventos usando seletores de eventos básicos](#) no Guia do AWS CloudTrail usuário.

- Para API obter detalhes, consulte [PutEventSelectors](#) na Referência de AWS CLI Comandos.

remove-tags

O código de exemplo a seguir mostra como usar `remove-tags`.

AWS CLI

Para remover etiquetas de uma trilha

O `remove-tags` comando a seguir remove as tags especificadas para `Trail1`:

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- Para API obter detalhes, consulte [RemoveTags](#) na Referência de AWS CLI Comandos.

start-logging

O código de exemplo a seguir mostra como usar `start-logging`.

AWS CLI

Para começar a registrar uma trilha

O `start-logging` comando a seguir ativa o registro em log para `Trail1`:

```
aws cloudtrail start-logging --name Trail1
```

- Para API obter detalhes, consulte [StartLogging](#) na Referência de AWS CLI Comandos.

stop-logging

O código de exemplo a seguir mostra como usar `stop-logging`.

AWS CLI

Para parar de registrar uma trilha

O `stop-logging` comando a seguir desativa o registro em log para `Trail1`:

```
aws cloudtrail stop-logging --name Trail1
```

- Para API obter detalhes, consulte [StopLogging](#) na Referência de AWS CLI Comandos.

update-subscription

O código de exemplo a seguir mostra como usar `update-subscription`.

AWS CLI

Para atualizar as configurações de uma trilha

O `update-subscription` comando a seguir atualiza a trilha para especificar um novo bucket e SNS tópico do S3:

```
aws cloudtrail update-subscription --name Trail1 --s3-new-bucket my-bucket-new --  
sns-new-topic my-topic-new
```

Saída:

```
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"  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [UpdateSubscription](#) na Referência de AWS CLI Comandos.

update-trail

O código de exemplo a seguir mostra como usar `update-trail`.

AWS CLI

Para atualizar uma trilha

O `update-trail` comando a seguir atualiza uma trilha para usar um bucket existente para entrega de registros:

```
aws cloudtrail update-trail --name Trail1 --s3-bucket-name my-bucket
```

Saída:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- Para API obter detalhes, consulte [UpdateTrail](#) na Referência de AWS CLI Comandos.

validate-logs

O código de exemplo a seguir mostra como usar `validate-logs`.

AWS CLI

Para validar um arquivo de log

O `validate-logs` comando a seguir valida os registros para `Trail1`:


```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

Saída:

```
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
```

- Para API obter detalhes, consulte [ValidateLogs](#) na Referência de AWS CLI Comandos.

CloudWatch exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CloudWatch.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-alarms

O código de exemplo a seguir mostra como usar `delete-alarms`.

AWS CLI

Como excluir um alarme

O exemplo a seguir usa o `delete-alarms` comando para excluir o CloudWatch alarme da Amazon chamado “myalarm”:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Saída:

```
This command returns to the prompt if successful.
```

- Para API obter detalhes, consulte [DeleteAlarms](#) na Referência de AWS CLI Comandos.

describe-alarm-history

O código de exemplo a seguir mostra como usar `describe-alarm-history`.

AWS CLI

Como recuperar o histórico de um alarme

O exemplo a seguir usa o `describe-alarm-history` comando para recuperar o histórico do CloudWatch alarme da Amazon chamado “myalarm”:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type StateUpdate
```

Saída:

```
{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeAlarmHistory](#) na Referência de AWS CLI Comandos.

describe-alarms-for-metric

O código de exemplo a seguir mostra como usar `describe-alarms-for-metric`.

AWS CLI

Como exibir informações sobre os alarmes associados a uma métrica

O exemplo a seguir usa o `describe-alarms-for-metric` comando para exibir informações sobre qualquer alarme associado à EC2 CPUUtilization métrica da Amazon e à instância com o ID `i-0c986c72`.

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Saída:

```

{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [DescribeAlarmsForMetric](#) na Referência de AWS CLI Comandos.

describe-alarms

O código de exemplo a seguir mostra como usar `describe-alarms`.

AWS CLI

Como listar informações sobre um alarme

O seguinte exemplo usa o comando `describe-alarms` para fornecer informações sobre o alarme chamado "myalarm":

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Saída:

```

{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeAlarms](#) na Referência de AWS CLI Comandos.

disable-alarm-actions

O código de exemplo a seguir mostra como usar `disable-alarm-actions`.

AWS CLI

Como desabilitar ações de um alarme

O seguinte exemplo usa o comando `disable-alarm-actions` para desabilitar todas as ações do alarme “myalarm”:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Esse comando retornará à solicitação, se houver êxito.

- Para API obter detalhes, consulte [DisableAlarmActions](#) na Referência de AWS CLI Comandos.

enable-alarm-actions

O código de exemplo a seguir mostra como usar `enable-alarm-actions`.

AWS CLI

Como habilitar todas as ações de um alarme

O seguinte exemplo usa o comando `enable-alarm-actions` para habilitar todas as ações para o alarme “myalarm”:

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Esse comando retornará à solicitação, se houver êxito.

- Para API obter detalhes, consulte [EnableAlarmActions](#) na Referência de AWS CLI Comandos.

get-metric-statistics

O código de exemplo a seguir mostra como usar `get-metric-statistics`.

AWS CLI

Para obter a CPU utilização por instância EC2

O exemplo a seguir usa o `get-metric-statistics` comando para obter a CPU utilização de uma EC2 instância com o 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
```

Saída:

```
{
  "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"
}
```

Especificar várias dimensões

O exemplo a seguir ilustra como especificar diversas dimensões. Cada dimensão é especificada como um par de nome/valor, com uma vírgula entre o nome e o valor. As diversas dimensões são separadas por um espaço. Se uma única métrica incluir diversas dimensões, você deverá especificar um valor para cada dimensão definida.

Para obter mais exemplos de uso do `get-metric-statistics` comando, consulte [Obter estatísticas para uma métrica no 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
```

- Para API obter detalhes, consulte [GetMetricStatistics](#) na Referência de AWS CLI Comandos.

list-metrics

O código de exemplo a seguir mostra como usar `list-metrics`.

AWS CLI

Para listar as métricas da Amazon SNS

O `list-metrics` exemplo a seguir exibe as métricas da AmazonSNS.

```
aws cloudwatch list-metrics \
--namespace "AWS/SNS"
```

Saída:

```
{
  "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"
  }
]
```

- Para API obter detalhes, consulte [ListMetrics](#) na Referência de AWS CLI Comandos.

put-metric-alarm

O código de exemplo a seguir mostra como usar `put-metric-alarm`.

AWS CLI

Para enviar uma mensagem de e-mail do Amazon Simple Notification Service quando CPU a utilização exceder 70 por cento

O exemplo a seguir usa o `put-metric-alarm` comando para enviar uma mensagem de e-mail do Amazon Simple CPU Notification Service quando a utilização excede 70%:

```
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
```

Esse comando retornará à solicitação, se houver êxito. Se já existir um alarme com o mesmo nome, ele será substituído pelo novo alarme.

Como especificar diversas dimensões

O exemplo a seguir ilustra como especificar diversas dimensões. Cada dimensão é especificada como um par de nome/valor, com uma vírgula entre o nome e o valor. As diversas dimensões são separadas por um espaço:

```
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
```

- Para API obter detalhes, consulte [PutMetricAlarm](#) na Referência de AWS CLI Comandos.

put-metric-data

O código de exemplo a seguir mostra como usar `put-metric-data`.

AWS CLI

Para publicar uma métrica personalizada na Amazon CloudWatch

O exemplo a seguir usa o `put-metric-data` comando para publicar uma métrica personalizada na Amazon CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://metric.json
```

Os valores da métrica em si são armazenados no JSON arquivo, `metric.json`.

Veja o conteúdo desse arquivo:

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Para obter mais informações, consulte [Publicação de métricas personalizadas no Amazon CloudWatch Developer Guide](#).

Como especificar diversas dimensões

O exemplo a seguir ilustra como especificar diversas dimensões. Cada dimensão é especificada como um par de nome/valor. As diversas dimensões são separadas por uma vírgula.

```
aws cloudwatch put-metric-data --metric-name Buffers --
namespace MyNameSpace --unit Bytes --value 231434333 --
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- Para API obter detalhes, consulte [PutMetricData](#) na Referência de AWS CLI Comandos.

set-alarm-state

O código de exemplo a seguir mostra como usar `set-alarm-state`.

AWS CLI

Para alterar temporariamente o estado de um alarme

O exemplo a seguir usa o `set-alarm-state` comando para alterar temporariamente o estado de um CloudWatch alarme da Amazon chamado “myalarm” e configurá-lo para o ALARM estado para fins de teste:

```
aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-
reason "testing purposes"
```

Esse comando retornará à solicitação, se houver êxito.

- Para API obter detalhes, consulte [SetAlarmState](#) na Referência de AWS CLI Comandos.

CloudWatch Exemplos de registros usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CloudWatch Logs.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-log-group

O código de exemplo a seguir mostra como usar `create-log-group`.

AWS CLI

O seguinte comando cria um grupo de logs chamado `my-logs`:

```
aws logs create-log-group --log-group-name my-logs
```

- Para API obter detalhes, consulte [CreateLogGroup](#) na Referência de AWS CLI Comandos.

create-log-stream

O código de exemplo a seguir mostra como usar `create-log-stream`.

AWS CLI

O seguinte comando cria um fluxo de logs `20150601` no grupo de logs `my-logs`:


```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Para API obter detalhes, consulte [CreateLogStream](#) na Referência de AWS CLI Comandos.

delete-log-group

O código de exemplo a seguir mostra como usar `delete-log-group`.

AWS CLI

O seguinte comando exclui um grupo de logs chamado `my-logs`:

```
aws logs delete-log-group --log-group-name my-logs
```

- Para API obter detalhes, consulte [DeleteLogGroup](#) na Referência de AWS CLI Comandos.

delete-log-stream

O código de exemplo a seguir mostra como usar `delete-log-stream`.

AWS CLI

O comando a seguir exclui um fluxo de registros chamado `20150531` de um grupo de registros chamado `my-logs`:

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- Para API obter detalhes, consulte [DeleteLogStream](#) na Referência de AWS CLI Comandos.

delete-retention-policy

O código de exemplo a seguir mostra como usar `delete-retention-policy`.

AWS CLI

O comando a seguir remove a política de retenção que foi aplicada anteriormente a um grupo de registros chamado `my-logs`:

```
aws logs delete-retention-policy --log-group-name my-logs
```

- Para API obter detalhes, consulte [DeleteRetentionPolicy](#) na Referência de AWS CLI Comandos.

describe-log-groups

O código de exemplo a seguir mostra como usar `describe-log-groups`.

AWS CLI

O seguinte comando descreve um grupo de logs chamado `my-logs`:

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

Saída:

```
{
  "logGroups": [
    {
      "storedBytes": 0,
      "metricFilterCount": 0,
      "creationTime": 1433189500783,
      "logGroupName": "my-logs",
      "retentionInDays": 5,
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLogGroups](#) na Referência de AWS CLI Comandos.

describe-log-streams

O código de exemplo a seguir mostra como usar `describe-log-streams`.

AWS CLI

O comando a seguir mostra todos os fluxos de log começando com o prefixo `2015` no grupo de registros: `my-logs`

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLogStreams](#) na Referência de AWS CLI Comandos.

get-log-events

O código de exemplo a seguir mostra como usar `get-log-events`.

AWS CLI

O comando a seguir recupera eventos de log de um stream de log chamado `20150601` no grupo `my-logs` de log:

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Saída:

```
{
  "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"
}

```

- Para API obter detalhes, consulte [GetLogEvents](#) na Referência de AWS CLI Comandos.

put-log-events

O código de exemplo a seguir mostra como usar `put-log-events`.

AWS CLI

O comando a seguir coloca eventos de log em um fluxo de log chamado `20150601` no grupo de registros `my-logs`:

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events
```

Saída:

```
{
  "nextSequenceToken": "49542672486831074009579604567656788214806863282469607346"
}
```

O exemplo acima lê uma JSON matriz de eventos de um arquivo chamado `events` no diretório atual:

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

Cada chamada subsequente exige que o próximo token de sequência fornecido pela chamada anterior seja especificado com a opção de token de sequência:

```
aws logs put-log-events --log-group-name my-logs --log-
stream-name 20150601 --log-events file://events2 --sequence-
token "49542672486831074009579604567656788214806863282469607346"
```

Saída:

```
{
  "nextSequenceToken": "49542672486831074009579604567900991230369019956308219826"
}
```

- Para API obter detalhes, consulte [PutLogEvents](#) na Referência de AWS CLI Comandos.

put-retention-policy

O código de exemplo a seguir mostra como usar `put-retention-policy`.

AWS CLI

O comando a seguir adiciona uma política de retenção de 5 dias a um grupo de registros chamados `my-logs`:

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- Para API obter detalhes, consulte [PutRetentionPolicy](#) na Referência de AWS CLI Comandos.

CloudWatch Exemplos de monitoramento de rede usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Monitoramento CloudWatch de Rede.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-monitor

O código de exemplo a seguir mostra como usar `create-monitor`.

AWS CLI

Exemplo 1: Para criar um monitor de rede com um período de agregação

O `create-monitor` exemplo a seguir cria um monitor chamado `Example_NetworkMonitor` com um `aggregationPeriod` conjunto de 30 segundos. A inicial state do monitor será `INACTIVE` porque não há sondas associadas a ele. O estado muda para `ACTIVE` somente quando as sondas são adicionadas. Você pode usar os comandos [update-monitor](#) ou [create-probe para adicionar sondas](#) a esse monitor.

```
aws networkmonitor create-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

Saída:

```
{
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
  "monitorName": "Example_NetworkMonitor",
  "state": "INACTIVE",
  "aggregationPeriod": 30,
  "tags": {}
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

Exemplo 2: Para criar um monitor de rede com uma sonda usando TCP e também incluindo tags

O `create-monitor` exemplo a seguir cria um monitor chamado `Example_NetworkMonitor`. O comando também cria uma sonda que usa o ICMP protocolo e inclui tags. Como não `aggregationPeriod` é passado na solicitação, 60 segundos são definidos como padrão. O `state` do monitor com a sonda permanecerá `PENDING` até que o monitor esteja `ACTIVE`. Isso pode levar alguns minutos, momento em que `state` mudará para `ACTIVE`, e você poderá começar a visualizar CloudWatch as métricas.

```
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
```

Saída:

```
{
  "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/
Example_NetworkMonitor",
  "monitorName": "Example_NetworkMonitor",
  "state": "PENDING",
  "aggregationPeriod": 60,
  "tags": {
    "Monitor": "Monitor1"
  }
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

Exemplo 3: Para criar um monitor de rede com uma sonda usando ICMP e também incluindo tags

O `create-monitor` exemplo a seguir cria um monitor chamado `Example_NetworkMonitor` com um `aggregationPeriod` de 30 segundos. O comando também cria uma sonda que usa o ICMP protocolo e inclui tags. Como não `aggregationPeriod` é passado na solicitação, 60 segundos são definidos como padrão. O `state` do monitor com a sonda permanecerá `PENDING` até que o monitor esteja `ACTIVE`. Isso pode levar alguns minutos, momento em que `state` mudará para `ACTIVE`, e você poderá começar a visualizar CloudWatch as métricas.

```
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
```

Saída:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [CreateMonitor](#) na Referência de AWS CLI Comandos.

create-probe

O código de exemplo a seguir mostra como usar `create-probe`.

AWS CLI

Exemplo 1: Para criar uma sonda que usa TCP e a adiciona a um monitor de rede

O `create-probe` exemplo a seguir cria uma sonda que usa TCP protocol e adiciona a sonda a um monitor chamado `Example_NetworkMonitor`. Uma vez criado, o state monitor com a sonda permanecerá `PENDING` até que o monitor esteja `ACTIVE`. Isso pode levar alguns minutos, momento em que o estado mudará para `ACTIVE`, e você poderá começar a visualizar CloudWatch as métricas.

```
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}
```

Saída:

```
{  
  "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"  
  }  
}
```

Exemplo 2: Para criar uma sonda que usa a sonda usando ICMP e adicioná-la a um monitor de rede

O `create-probe` exemplo a seguir cria uma sonda que usa ICMP protocol e adiciona a sonda a um monitor chamado `Example_NetworkMonitor`. Uma vez criado, o state monitor com a sonda permanecerá `PENDING` até que o monitor esteja `ACTIVE`. Isso pode levar alguns minutos, momento em que o estado mudará para `ACTIVE`, e você poderá começar a visualizar CloudWatch as métricas.

```
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}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [CreateProbe](#) na Referência de AWS CLI Comandos.

delete-monitor

O código de exemplo a seguir mostra como usar delete-monitor.

AWS CLI

Para excluir um monitor

O delete-monitor exemplo a seguir exclui um monitor chamado `Example_NetworkMonitor`.

```
aws networkmonitor delete-monitor \  
  --monitor-name Example_NetworkMonitor
```

Este comando não produz saída.

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteMonitor](#) na Referência de AWS CLI Comandos.

delete-probe

O código de exemplo a seguir mostra como usar delete-probe.

AWS CLI

Para excluir uma sonda

O delete-probe exemplo a seguir exclui uma sonda com o ID probe-12345 de um monitor de rede chamado Example_NetworkMonitor

```
aws networkmonitor delete-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345
```

Este comando não produz saída.

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteProbe](#) na Referência de AWS CLI Comandos.

get-monitor

O código de exemplo a seguir mostra como usar get-monitor.

AWS CLI

Para obter informações sobre o monitor

O get-monitor exemplo a seguir obtém informações sobre um monitor chamado Example_NetworkMonitor.

```
aws networkmonitor get-monitor \  
  --monitor-name Example_NetworkMonitor
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [GetMonitor](#) na Referência de AWS CLI Comandos.

get-probe

O código de exemplo a seguir mostra como usar `get-probe`.

AWS CLI

Para visualizar os detalhes da sonda

O `get-probe` exemplo a seguir retorna detalhes sobre uma sonda associada a `probeID` `probe-12345` um monitor chamado `Example_NetworkMonitor`.

```
aws networkmonitor get-probe \
  --monitor-name Example_NetworkMonitor \
  --probe-id probe-12345
```

Saída:

```
{
  "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"  
}  
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [GetProben](#)a Referência de AWS CLI Comandos.

list-monitors

O código de exemplo a seguir mostra como usar `list-monitors`.

AWS CLI

Exemplo 1: Para listar todos os monitores (monitor único)

O `list-monitors` exemplo a seguir retorna uma lista de apenas um único monitor. O monitor `state` é `ACTIVE` e tem um `aggregationPeriod` de 60 segundos.

```
aws networkmonitor list-monitors
```

Saída:

```
{  
  "monitors": [{  
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
    "monitorName": "Example_NetworkMonitor",  
    "state": "ACTIVE",  
    "aggregationPeriod": 60,  
    "tags": {  
      "Monitor": "Monitor1"  
    }  
  }  
}
```

```
    }  
  }  
]  
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

Exemplo 2: Para listar todos os monitores (vários monitores)

O `list-monitors` exemplo a seguir retorna uma lista de três monitores. O `state` único monitor é `ACTIVE` e gera CloudWatch métricas. Os estados dos outros dois monitores estão `INACTIVE` gerando CloudWatch métricas. Todos os três monitores usam um `aggregationPeriod` de 60 segundos.

```
aws networkmonitor list-monitors
```

Saída:

```
{  
  "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": {}
    }
]
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [ListMonitors](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir retorna uma lista das tags de um monitor chamado `Example_NetworkMonitor`.

```
aws networkmonitor list-tags-for-resource \
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor
```

Saída:

```
{
  "tags": {
    "Environment": "Dev",
    "Application": "PetStore"
  }
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir marca um monitor chamado `Example_NetworkMonitor` com `Environment=Dev` e `Application=PetStore` tags.

```
aws networkmonitor tag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
  --tags Environment=Dev,Application=PetStore
```

Este comando não produz saída.

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para desmarcar um recurso

O `untag-resource` exemplo a seguir remove um `tag-keys` parâmetro com o par de valores-chave `Environment` `Application` de sua associação com um monitor chamado `Example_NetworkMonitor`

```
aws networkmonitor untag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
  --tag-keys Environment Application
```

Este comando não produz saída.

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-monitor

O código de exemplo a seguir mostra como usar update-monitor.

AWS CLI

Para atualizar um monitor

O update-monitor exemplo a seguir altera o monitor aggregationPeriod de 60 segundos para 30 segundos.

```
aws networkmonitor update-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

Saída:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateMonitor](#) na Referência de AWS CLI Comandos.

update-probe

O código de exemplo a seguir mostra como usar update-probe.

AWS CLI

Para atualizar uma sonda

O `update-probe` exemplo a seguir atualiza o endereço `destination` IP original de uma sonda e também atualiza o `packetSize` para `60`.

```
aws networkmonitor update-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345 \  
  --destination 10.0.0.150 \  
  --packet-size 60
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Como o Amazon CloudWatch Network Monitor funciona](#) no Guia CloudWatch do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateProbe](#) na Referência de AWS CLI Comandos.

CodeArtifact exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodeArtifact.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-external-connection

O código de exemplo a seguir mostra como usar `associate-external-connection`.

AWS CLI

Para adicionar uma conexão externa a um repositório

O `associate-external-connection` exemplo a seguir adiciona uma conexão externa ao `npmjs.com` a um repositório chamado `test-repo`.

```
aws codeartifact associate-external-connection \  
  --repository test-repo \  
  --domain test-domain \  
  --external-connection public:npmjs
```

Saída:

```
{  
  "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"
    }
]
}
}

```

Para obter mais informações, consulte [Adicionar uma conexão externa](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [AssociateExternalConnection](#) na Referência de AWS CLI Comandos.

copy-package-versions

O código de exemplo a seguir mostra como usar `copy-package-versions`.

AWS CLI

Para copiar versões do pacote de um repositório para outro

O seguinte `copy-package-versions` move as versões 4.0.0 e 5.0.0 de um pacote chamado `test-package` de `my-repo` para `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"]'

```

Saída:

```

{
  "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"
  }
]
```

Para obter mais informações, consulte [Copiar pacotes entre repositórios](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [CopyPackageVersions](#) na Referência de AWS CLI Comandos.

create-domain

O código de exemplo a seguir mostra como usar `create-domain`.

AWS CLI

Para criar um domínio

O `create-domain` exemplo a seguir cria um domínio chamado `test-domain`.

```
aws codeartifact create-domain \
  --domain test-domain
```

Saída:

```
{
  "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
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Criar um domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [CreateDomain](#) na Referência de AWS CLI Comandos.

create-repository

O código de exemplo a seguir mostra como usar `create-repository`.

AWS CLI

Para criar um repositório

O `create-repository` exemplo a seguir cria um repositório chamado `test-repo` dentro de um domínio chamado `test-domain`.

```
aws codeartifact create-repository \  
  --domain test-domain \  
  --domain-owner 111122223333 \  
  --repository test-repo \  
  --description "This is a test repository."
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Criar um domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [CreateRepository](#) na Referência de AWS CLI Comandos.

delete-domain-permissions-policy

O código de exemplo a seguir mostra como usar `delete-domain-permissions-policy`.

AWS CLI

Para excluir o documento de política de permissões de um domínio

O `delete-domain-permissions-policy` exemplo a seguir exclui a política de permissão de um domínio chamado `test-domain`.

```
aws codeartifact delete-domain-permissions-policy \  
  --domain test-domain
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Excluir uma política de domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DeleteDomainPermissionsPolicy](#) na Referência de AWS CLI Comandos.

delete-domain

O código de exemplo a seguir mostra como usar `delete-domain`.

AWS CLI

Para excluir um domínio

O `delete-domain` exemplo a seguir exclui um domínio chamado `test-domain`.

```
aws codeartifact delete-domain \  
  --domain test-domain
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Excluir um domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DeleteDomain](#) na Referência de AWS CLI Comandos.

delete-package-versions

O código de exemplo a seguir mostra como usar `delete-package-versions`.

AWS CLI

Para excluir versões do pacote

O `delete-package-versions` exemplo a seguir exclui a versão 4.0.0 de um pacote chamado `test-package`.


```
aws codeartifact delete-package-versions \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0
```

Saída:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Deleted"  
    }  
  },  
  "failedVersions": {}  
}
```

Para obter mais informações, consulte [Excluir uma versão do pacote](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DeletePackageVersions](#) na Referência de AWS CLI Comandos.

delete-repository-permissions-policy

O código de exemplo a seguir mostra como usar `delete-repository-permissions-policy`.

AWS CLI

Para excluir uma política de permissões de um repositório

O `delete-repository-permissions-policy` exemplo a seguir exclui a política de permissão de um repositório chamado `test-repo`.

```
aws codeartifact delete-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo
```

Saída:

```
{
  "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": "*"
    }
  ]
}
```

Para obter mais informações, consulte [Excluir uma política](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DeleteRepositoryPermissionsPolicy](#) na Referência de AWS CLI Comandos.

delete-repository

O código de exemplo a seguir mostra como usar `delete-repository`.

AWS CLI

Para excluir um repositório

O `delete-repository` exemplo a seguir exclui um repositório chamado `test-repo` em um domínio chamado `test-domain`

```
aws codeartifact delete-repository \
  --domain test-domain \
```

```
--repository test-repo
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Excluir um repositório](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [DeleteRepository](#) na Referência de AWS CLI Comandos.

describe-domain

O código de exemplo a seguir mostra como usar `describe-domain`.

AWS CLI

Para obter informações sobre um domínio

O `describe-domain` exemplo a seguir retorna um `DomainDescription` objeto para um domínio chamado `test-domain`.

```
aws codeartifact describe-domain \  
  --domain test-domain
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Visão geral do domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DescribeDomain](#) na Referência de AWS CLI Comandos.

describe-repository

O código de exemplo a seguir mostra como usar `describe-repository`.

AWS CLI

Para obter informações sobre um repositório

O `describe-repository` exemplo a seguir retorna um `RepositoryDescription` objeto para um repositório chamado `test-repo`.

```
aws codeartifact describe-repository \
  --domain test-domain \
  --repository test-repo
```

Saída:

```
{
  "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": []
  }
}

```

Para obter mais informações, consulte [Criar um domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DescribeRepository](#) na Referência de AWS CLI Comandos.

disassociate-external-connection

O código de exemplo a seguir mostra como usar `disassociate-external-connection`.

AWS CLI

Para remover uma conexão externa de um repositório

O `disassociate-external-connection` exemplo a seguir remove uma conexão externa com `npmjs.com` de um repositório chamado `test-repo`.

```

aws codeartifact disassociate-external-connection \
  --repository test-repo \
  --domain test-domain \
  --external-connection public:npmjs

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Remover uma conexão externa](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DisassociateExternalConnection](#) na Referência de AWS CLI Comandos.

dispose-package-versions

O código de exemplo a seguir mostra como usar `dispose-package-versions`.

AWS CLI

Para excluir os ativos de uma versão do pacote e definir seu status como Descartado

O `dispose-package-versions` exemplo a seguir exclui os ativos do pacote de teste versão 4.0.0 e define seu status como Disposed.

```
aws codeartifact dispose-package-versions \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0
```

Saída:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Disposed"  
    }  
  },  
  "failedVersions": {}  
}
```

Para obter mais informações, consulte Como [trabalhar com pacotes CodeArtifact no](#) Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [DisposePackageVersions](#) na Referência de AWS CLI Comandos.

get-authorization-token

O código de exemplo a seguir mostra como usar `get-authorization-token`.

AWS CLI

Para obter um token de autorização

O `get-authorization-token` exemplo a seguir recupera um token de CodeArtifact autorização.

```
aws codeartifact get-authorization-token \  
  --domain test-domain \  
  --query authorizationToken \  
  --output text
```

Saída:

```
This command will return the authorization token. You can store the output in an  
environment variable when calling the command.
```

Para obter mais informações, consulte [Configurar o pip sem o comando de login](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [GetAuthorizationToken](#) na Referência de AWS CLI Comandos.

get-domain-permissions-policy

O código de exemplo a seguir mostra como usar `get-domain-permissions-policy`.

AWS CLI

Para obter o documento de política de permissões para um domínio

O `get-domain-permissions-policy` exemplo a seguir anexa a política de permissão a um domínio chamado `test-domain`.

```
aws codeartifact get-domain-permissions-policy \  
  --domain test-domain
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Leia uma política de domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [GetDomainPermissionsPolicy](#) na Referência de AWS CLI Comandos.

get-package-version-asset

O código de exemplo a seguir mostra como usar `get-package-version-asset`.

AWS CLI

Para obter um ativo de uma versão do pacote

O `get-package-version-asset` exemplo a seguir recupera o `package.tgz` ativo para a versão 4.0.0 de um pacote npm chamado `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
```

Saída:

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=" }  
}
```

Para obter mais informações, consulte [Listar ativos da versão do pacote](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [GetPackageVersionAsset](#) na Referência de AWS CLI Comandos.

get-package-version-readme

O código de exemplo a seguir mostra como usar get-package-version-readme.

AWS CLI

Para obter o arquivo readme de uma versão do pacote

O get-package-version-readme exemplo a seguir recupera o arquivo readme da versão 4.0.0 de um pacote npm chamado test-package.

```
aws codeartifact get-package-version-readme \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

Saída:

```
{
```

```
"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="
}
```

Para obter mais informações, consulte [Exibir arquivo readme da versão do pacote](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [GetPackageVersionReadme](#) na Referência de AWS CLI Comandos.

get-repository-endpoint

O código de exemplo a seguir mostra como usar `get-repository-endpoint`.

AWS CLI

Para obter o endpoint de um repositório URL

O `get-repository-endpoint` exemplo a seguir retorna o endpoint npm para o repositório `test-repo`.

```
aws codeartifact get-repository-endpoint \
  --domain test-domain \
  --repository test-repo \
  --format npm
```

Saída:

```
{
  "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-west-2.amazonaws.com/npm/test-repo/"
}
```

Para obter mais informações, consulte [Conectar-se a um repositório](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [GetRepositoryEndpoint](#) na Referência de AWS CLI Comandos.

get-repository-permissions-policy

O código de exemplo a seguir mostra como usar `get-repository-permissions-policy`.

AWS CLI

Para obter o documento de política de permissões para um repositório

O `get-repository-permissions-policy` exemplo a seguir anexa a política de permissão a um repositório chamado `test-repo`.

```
aws codeartifact get-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo
```

Saída:

```
{  
  "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": "*"   
    }  
  ]  
}
```

Para obter mais informações, consulte [Leia uma política](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [GetRepositoryPermissionsPolicy](#) na Referência de AWS CLI Comandos.

list-domains

O código de exemplo a seguir mostra como usar `list-domains`.

AWS CLI

Para listar domínios

O `list-domains` exemplo a seguir retorna um resumo de todos os domínios pertencentes à AWS conta que faz a chamada.

```
aws codeartifact list-domains
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com domínios CodeArtifact no](#) Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [ListDomains](#) na Referência de AWS CLI Comandos.

list-package-version-assets

O código de exemplo a seguir mostra como usar `list-package-version-assets`.

AWS CLI

Para visualizar os ativos de uma versão do pacote

O `list-package-version-assets` exemplo a seguir recupera os ativos da versão 4.0.0 de um pacote npm chamado `test-package`.

```
aws codeartifact list-package-version-assets \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --package-version 4.0.0
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Listar ativos da versão do pacote](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [ListPackageVersionAssets](#) na Referência de AWS CLI Comandos.

list-package-version-dependencies

O código de exemplo a seguir mostra como usar `list-package-version-dependencies`.

AWS CLI

Para ver as dependências da versão de um pacote

O `list-package-version-dependencies` exemplo a seguir recupera as dependências da versão 4.0.0 de um pacote npm chamado `test-package`.

```
aws codeartifact list-package-version-dependencies \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
}
```

Para obter mais informações, consulte [Exibir e atualizar os detalhes e dependências da versão do pacote](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [ListPackageVersionDependencies](#) na Referência de AWS CLI Comandos.

list-package-versions

O código de exemplo a seguir mostra como usar `list-package-versions`.

AWS CLI

Para listar as versões de um pacote

O `list-package-versions` exemplo a seguir retorna uma lista de versões de pacotes para um pacote chamado `kind-of`.

```
aws codeartifact list-package-versions \  
  --package kind-of \  
  --domain test-domain \  
  --repository test-repo \  
  --format npm
```

Saída:

```
{  
  "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"
  }
]
}
```

Para obter mais informações, consulte [Listar versões do pacote](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [ListPackageVersions](#) na Referência de AWS CLI Comandos.

list-packages

O código de exemplo a seguir mostra como usar `list-packages`.

AWS CLI

Para listar pacotes em um repositório

O `list-packages` exemplo a seguir lista pacotes em um repositório chamado `test-repo` em um domínio chamado `test-domain`.

```
aws codeartifact list-packages \
  --domain test-domain \
  --repository test-repo
```

Saída:

```
{
  "packages": [
    {
      "format": "npm",
```



```
        "package": "lodash"
      }
    {
      "format": "python",
      "package": "test-package"
    }
  ]
}
```

Para obter mais informações, consulte [Listar nomes de pacotes](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [ListPackages](#) na Referência de AWS CLI Comandos.

list-repositories-in-domain

O código de exemplo a seguir mostra como usar `list-repositories-in-domain`.

AWS CLI

Para listar repositórios em um domínio

O `list-repositories-in-domain` exemplo a seguir retorna um resumo de todos os repositórios no domínio `test-domain`.

```
aws codeartifact list-repositories-in-domain \
  --domain test-domain
```

Saída:

```
{
  "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."
  }
]
}
```

Para obter mais informações, consulte [Listar repositórios](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [ListRepositoriesInDomain](#) na Referência de AWS CLI Comandos.

list-repositories

O código de exemplo a seguir mostra como usar `list-repositories`.

AWS CLI

Para listar repositórios

O `list-repositories` exemplo a seguir retorna um resumo de todos os repositórios no domínio de propriedade da AWS conta que faz a chamada.

```
aws codeartifact list-repositories
```

Saída:

```
{
  "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."
    }
]
}

```

Para obter mais informações, consulte [Listar repositórios](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [ListRepositories](#) na Referência de AWS CLI Comandos.

login

O código de exemplo a seguir mostra como usar login.

AWS CLI

Para configurar a autenticação no seu repositório com o comando login

O login exemplo a seguir configura o gerenciador de pacotes npm com um repositório chamado test-repo em um domínio chamado test-domain.

```

aws codeartifact login \
  --domain test-domain \
  --repository test-repo \
  --tool npm

```

Saída:

```

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
```

Para obter mais informações, consulte [Introdução ao AWS CLI](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte Referência de AWS CLI comando de [login](#).

put-domain-permissions-policy

O código de exemplo a seguir mostra como usar `put-domain-permissions-policy`.

AWS CLI

Para anexar uma política de permissões a um domínio

O `put-domain-permissions-policy` exemplo a seguir anexa uma política de permissão definida no arquivo `policy.json` a um domínio chamado `test-domain`.

```
aws codeartifact put-domain-permissions-policy \
  --domain test-domain \
  --policy-document file://PATH/T0/policy.json
```

Saída:

```
{
  "policy": {
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-
domain",
    "document": "{ ...policy document content...}",
    "revision": "MQLyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxxx="
  }
}
```

Para obter mais informações, consulte [Definir uma política de domínio](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [PutDomainPermissionsPolicy](#) na Referência de AWS CLI Comandos.

put-repository-permissions-policy

O código de exemplo a seguir mostra como usar `put-repository-permissions-policy`.

AWS CLI

Para anexar uma política de permissões a um repositório

O `put-repository-permissions-policy` exemplo a seguir anexa uma política de permissão definida no arquivo `policy.json` a um repositório chamado `test-repo`.

```
aws codeartifact put-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo \  
  --policy-document file://PATH/T0/policy.json
```

Saída:

```
{  
  "policy": {  
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-domain/test-repo",  
    "document": "{ ...policy document content...}",  
    "revision": "MQ1yyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxxx="  }  
}
```

Para obter mais informações, consulte [Definir uma política](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [PutRepositoryPermissionsPolicy](#) na Referência de AWS CLI Comandos.

update-package-versions-status

O código de exemplo a seguir mostra como usar `update-package-versions-status`.

AWS CLI

Para atualizar o status da versão do pacote

O `update-package-versions-status` exemplo a seguir atualiza o status da versão 4.0.0 do pacote de teste para Arquivado.

```
aws codeartifact update-package-versions-status \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --version 4.0.0
```

```
--package test-package \  
--versions 4.0.0 \  
--target-status Archived
```

Saída:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Archived"  
    }  
  },  
  "failedVersions": {}  
}
```

Para obter mais informações, consulte [Atualizar o status da versão do pacote](#) no Guia AWS CodeArtifact do usuário.

- Para API obter detalhes, consulte [UpdatePackageVersionsStatus](#) na Referência de AWS CLI Comandos.

update-repository

O código de exemplo a seguir mostra como usar `update-repository`.

AWS CLI

Para atualizar um repositório

O `update-repository` exemplo a seguir atualiza a descrição de um repositório chamado `test-repo` em um domínio chamado `test-domain` para “esta é uma descrição atualizada”.

```
aws codeartifact update-repository \  
--domain test-domain \  
--repository test-repo \  
--description "this is an updated description"
```

Saída:

```
{  
  "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": []
  }
}
```

Para obter mais informações, consulte [Exibir ou modificar a configuração de um repositório](#) no Guia do AWS CodeArtifact usuário.

- Para API obter detalhes, consulte [UpdateRepository](#) na Referência de AWS CLI Comandos.

CodeBuild exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodeBuild.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-delete-builds

O código de exemplo a seguir mostra como usar batch-delete-builds.

AWS CLI

Para excluir compilações. AWS CodeBuild

O `batch-delete-builds` exemplo a seguir exclui compilações CodeBuild com o especificado. 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
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Excluir compilações \(AWS CLI\)](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [BatchDeleteBuilds](#) na Referência de AWS CLI Comandos.

`batch-get-build-batches`

O código de exemplo a seguir mostra como usar `batch-get-build-batches`.

AWS CLI

Para ver detalhes dos builds in AWS CodeBuild.

O `batch-get-build-batches` exemplo a seguir obtém informações sobre lotes de compilação CodeBuild com o especificado IDs.

```
aws codebuild batch-get-build-batches \
  --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```


Saída:

```
{
  "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": []
}

```

Para obter mais informações, consulte Batch builds em AWS CodeBuild < <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>> __ no Guia do usuário.AWS CodeBuild

- Para API obter detalhes, consulte [BatchGetBuildBatches](#) na Referência de AWS CLI Comandos.

batch-get-builds

O código de exemplo a seguir mostra como usar batch-get-builds.

AWS CLI

Para ver detalhes dos builds in AWS CodeBuild.

O batch-get-builds exemplo a seguir obtém informações sobre compilações CodeBuild com o especificado IDs.

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

Saída:

```
{
  "buildsNotFound": [],
  "builds": [
    {
      "artifacts": {
        "md5sum": "0e95edf915048a0c22efe6d139fff837",
        "location": "arn:aws:s3:::codepipeline-us-west-2-820783811474/CodeBuild-Python-Pip/BuildArtif/6DJsQa",
        "encryptionDisabled": false,
        "sha256sum":
"cfaf0df33a090966a737f64ae4fe498969fdc842a0c9aec540bf93c37ac0d05a2"
      },
      "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"
  }
]
}

```

Para obter mais informações, consulte [Exibir detalhes da compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.


```
        "type": "NO_ARTIFACTS"
      },
      "badge": {
        "badgeEnabled": false
      },
      "lastModified": 1540588091.108,
      "created": 1540588091.108,
      "arn": "arn:aws:codebuild:us-west-2:123456789012:project/test-for-
sample",
      "secondarySources": [],
      "secondaryArtifacts": [],
      "cache": {
        "type": "NO_CACHE"
      },
      "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-test-
role",
      "environment": {
        "image": "aws/codebuild/java:openjdk-8",
        "privilegedMode": true,
        "type": "LINUX_CONTAINER",
        "computeType": "BUILD_GENERAL1_SMALL",
        "environmentVariables": []
      },
      "tags": []
    },
    {
      "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
      "name": "my-other-demo-project",
      "queuedTimeoutInMinutes": 480,
      "timeoutInMinutes": 60,
      "source": {
        "location": "https://github.com/iversonic/codedeploy-sample.git",
        "reportBuildStatus": false,
        "buildspec": "buildspec.yml",
        "insecureSsl": false,
        "gitCloneDepth": 1,
        "type": "GITHUB",
        "auth": {
          "type": "OAUTH"
        }
      },
      "artifacts": {
        "type": "NO_ARTIFACTS"
      }
    }
  ],
  "tags": []
}
```

```

    "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": []
  }
]
}

```

Para obter mais informações, consulte [Exibir os detalhes de um projeto de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [BatchGetProjects](#) na Referência de AWS CLI Comandos.

batch-get-report-groups

O código de exemplo a seguir mostra como usar batch-get-report-groups.

AWS CLI

Para obter informações sobre um ou mais grupos de relatórios em AWS CodeBuild.

O batch-get-report-groups exemplo a seguir recupera informações sobre o grupo de relatórios com o especificado ARN.

```

aws codebuild batch-get-report-groups \
  --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>

```


Saída:

```
{
  "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": []
}
```

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [BatchGetReportGroups](#) na Referência de AWS CLI Comandos.

batch-get-reports

O código de exemplo a seguir mostra como usar batch-get-reports.

AWS CLI

Para obter informações sobre um ou mais relatórios em AWS CodeBuild.

O batch-get-reports exemplo a seguir recupera informações sobre os relatórios com o especificado 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>
```

Saída:

```
{
  "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": []
}

```

Para obter mais informações, consulte Como [trabalhar com relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [BatchGetReports](#) na Referência de AWS CLI Comandos.

create-project

O código de exemplo a seguir mostra como usar `create-project`.

AWS CLI

Exemplo 1: Para criar um projeto de AWS CodeBuild construção

O `create-project` exemplo a seguir cria um projeto de CodeBuild compilação usando arquivos de origem de um bucket do 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\"
```

Saída:

```
{
  "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
    }
  }
```

Exemplo 2: Para criar um projeto de AWS CodeBuild construção usando um arquivo JSON de entrada para os parâmetros

O `create-project` exemplo a seguir cria um projeto de CodeBuild construção passando todos os parâmetros necessários em um arquivo JSON de entrada. Crie o modelo do arquivo de entrada executando o comando somente com o `--generate-cli-skeleton` parameter.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

O JSON arquivo de entrada `create-project.json` contém o seguinte conteúdo:

```
{
  "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"
}
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Criar um projeto de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [CreateProject](#) na Referência de AWS CLI Comandos.

create-report-group

O código de exemplo a seguir mostra como usar `create-report-group`.

AWS CLI

Para criar um grupo de relatórios em AWS CodeBuild.

O `create-report-group` exemplo a seguir cria um novo grupo de relatórios.

```
aws codebuild create-report-group \
```

```
--cli-input-json file://create-report-group-source.json
```

Conteúdo create-report-group-source do.json:

```
{
  "name": "cli-created-report-group",
  "type": "TEST",
  "exportConfig": {
    "exportConfigType": "S3",
    "s3Destination": {
      "bucket": "my-s3-bucket",
      "path": "",
      "packaging": "ZIP",
      "encryptionDisabled": true
    }
  }
}
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [CreateReportGroup](#) na Referência de AWS CLI Comandos.

create-webhook

O código de exemplo a seguir mostra como usar create-webhook.

AWS CLI

Para criar filtros de webhook para um projeto AWS CodeBuild

O create-webhook exemplo a seguir cria um webhook para um CodeBuild projeto chamado my-project que tem dois grupos de filtros. O primeiro grupo de filtros especifica as solicitações pull que são criadas, atualizadas ou reabertas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/master$` e referências head que correspondam a `^refs/heads/myBranch$`. O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que não correspondem à expressão regular. `^refs/heads/myBranch$`

```
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}]]"
```

Saída:

```
{
  "webhook": {
    "payloadUrl": "https://codebuild.us-west-2.amazonaws.com/webhooks?
    t=eyJlbmNyeXB0ZWREYXRhIjoiVVV1MGtoeGRwSzZFRXl2Wnh4bld1Z0tKZ291TVpQNEtFamQ3RDlDYWpRaGIreVFrdm
    "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
    }
]
]
}

```

Para obter mais informações, consulte [Filtrar eventos de GitHub webhook \(SDK\)](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [CreateWebhook](#) na Referência de AWS CLI Comandos.

delete-build-batch

O código de exemplo a seguir mostra como usar delete-build-batch.

AWS CLI

Para excluir um lote incorporado AWS CodeBuild.

O delete-build-batch exemplo a seguir exclui a compilação em lote especificada.

```
aws codebuild delete-build-batch \  
  --id <project-name>:<batch-ID>
```

Saída:

```
{  
  "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": []  
}
```

Para obter mais informações, consulte [Batch builds in AWS CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [DeleteBuildBatch](#) na Referência de AWS CLI Comandos.

delete-project

O código de exemplo a seguir mostra como usar delete-project.

AWS CLI

Para excluir um projeto de AWS CodeBuild compilação

O delete-project exemplo a seguir exclui o projeto de CodeBuild construção especificado.

```
aws codebuild delete-project --name my-project
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um projeto de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [DeleteProject](#) na Referência de AWS CLI Comandos.

delete-report-group

O código de exemplo a seguir mostra como usar delete-report-group.

AWS CLI

Para excluir um grupo de relatórios em AWS CodeBuild.

O delete-report-group exemplo a seguir exclui o grupo de relatórios com o especificadoARN.

```
aws codebuild delete-report-group \  
  --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [DeleteReportGroup](#) na Referência de AWS CLI Comandos.

delete-report

O código de exemplo a seguir mostra como usar delete-report.

AWS CLI

Para excluir um relatório em AWS CodeBuild.

O delete-report exemplo a seguir exclui o relatório especificado.

```
aws codebuild delete-report \  
  --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [DeleteReport](#) na Referência de AWS CLI Comandos.

delete-source-credentials

O código de exemplo a seguir mostra como usar `delete-source-credentials`.

AWS CLI

Para se desconectar de um provedor de origem e remover seus tokens de acesso.

O `delete-source-credentials` exemplo a seguir se desconecta de um provedor de origem e remove seus tokens. As credenciais ARN de origem usadas para se conectar ao provedor de origem determinam quais credenciais de origem.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Saída:

```
{
  "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"
}
```

Para obter mais informações, consulte [Connect Source Providers with Access Tokens \(CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [DeleteSourceCredentials](#) na Referência de AWS CLI Comandos.

delete-webhook

O código de exemplo a seguir mostra como usar `delete-webhook`.

AWS CLI

Para excluir um filtro de webhook de um projeto AWS CodeBuild

O `delete-webhook` exemplo a seguir exclui um webhook do projeto especificado CodeBuild .

```
aws codebuild delete-webhook --project-name my-project
```

Este comando não produz saída.

Para obter mais informações, consulte [Parar de executar compilações automaticamente \(AWS CLI\)](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [DeleteWebhook](#) na Referência de AWS CLI Comandos.

describe-code-coverages

O código de exemplo a seguir mostra como usar `describe-code-coverages`.

AWS CLI

Para obter informações detalhadas sobre os resultados do teste de cobertura de código em AWS CodeBuild.

O `describe-code-coverages` exemplo a seguir obtém informações sobre os resultados do teste de cobertura de código no relatório especificado.

```
aws codebuild describe-code-coverages \  
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Relatórios de cobertura de código](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [DescribeCodeCoverages](#) na Referência de AWS CLI Comandos.

describe-test-cases

O código de exemplo a seguir mostra como usar `describe-test-cases`.

AWS CLI

Para obter informações detalhadas sobre casos de teste em AWS CodeBuild.

O `describe-test-cases` exemplo a seguir obtém informações sobre os casos de teste no relatório especificado.

```

aws codebuild describe-test-cases \
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>

```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte Como [trabalhar com relatórios de teste AWS CodeBuild no Guia AWS CodeBuild do usuário](#).

- Para API obter detalhes, consulte [DescribeTestCases](#) na Referência de AWS CLI Comandos.

import-source-credentials

O código de exemplo a seguir mostra como usar `import-source-credentials`.

AWS CLI

Conecte um AWS CodeBuild usuário a um provedor de origem importando as credenciais do provedor de origem.

O `import-source-credentials` exemplo a seguir importa um token para um repositório do Bitbucket que usa BASIC _ como seu AUTH tipo de autenticação.

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-
type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

Saída:

```
{
  "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"
```

```
}
```

Para obter mais informações, consulte [Connect Source Providers with Access Tokens \(CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ImportSourceCredentials](#) na Referência de AWS CLI Comandos.

invalidate-project-cache

O código de exemplo a seguir mostra como usar `invalidate-project-cache`.

AWS CLI

Para redefinir o cache de um projeto de AWS CodeBuild construção.

O `invalidate-project-cache` exemplo a seguir redefine o cache do CodeBuild projeto especificado.

```
aws codebuild invalidate-project-cache --project-name my-project
```

Este comando não produz saída.

Para obter mais informações, consulte [Build Caching in CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [InvalidateProjectCache](#) na Referência de AWS CLI Comandos.

list-build-batches-for-project

O código de exemplo a seguir mostra como usar `list-build-batches-for-project`.

AWS CLI

Para listar compilações em lote para um projeto de compilação específico em AWS CodeBuild.

O `list-build-batches-for-project` exemplo a seguir lista as compilações CodeBuild em lote para o projeto especificado.

```
aws codebuild list-build-batches-for-project \  
  --project-name "<project-name>"
```


Saída:

```
{
  "ids": [
    "<project-name>:<batch-ID>",
    "<project-name>:<batch-ID>"
  ]
}
```

Para obter mais informações, consulte [Batch builds in AWS CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [ListBuildBatchesForProject](#) na Referência de AWS CLI Comandos.

list-build-batches

O código de exemplo a seguir mostra como usar `list-build-batches`.

AWS CLI

Para listar compilações em AWS CodeBuild lote.

O `list-build-batches` exemplo a seguir lista as compilações CodeBuild em lote para a conta atual.

```
aws codebuild list-build-batches
```

Saída:

```
{
  "ids": [
    "<project-name>:<batch-ID>",
    "<project-name>:<batch-ID>"
  ]
}
```

Para obter mais informações, consulte [Batch builds em AWS CodeBuild < https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html >](https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html) no Guia do usuário.AWS CodeBuild

- Para API obter detalhes, consulte [ListBuildBatches](#) na Referência de AWS CLI Comandos.

list-builds-for-project

O código de exemplo a seguir mostra como usar `list-builds-for-project`.

AWS CLI

Para ver uma lista de compilações para um projeto de AWS CodeBuild compilação.

O `list-builds-for-project` exemplo a seguir lista a compilação IDs em ordem decrescente para o projeto de CodeBuild compilação especificado.

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order DESCENDING
```

Saída:

```
{
  "ids": [
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-11111example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-22222example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-33333example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-44444example",
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-55555example"
  ]
}
```

Para obter mais informações, consulte [Exibir uma lista de compilações IDs para um projeto de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário

- Para API obter detalhes, consulte [ListBuildsForProject](#) na Referência de AWS CLI Comandos.

list-builds

O código de exemplo a seguir mostra como usar `list-builds`.

AWS CLI

Para obter uma lista de AWS CodeBuild construções. IDs

O `list-builds` exemplo a seguir obtém uma lista de CodeBuild IDs classificados em ordem crescente.

```
aws codebuild list-builds --sort-order ASCENDING
```

A saída inclui um `nextToken` valor que indica que há mais saída disponível.

```
{
  "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"
  ]
}
```

Execute esse comando novamente e forneça o `nextToken` valor na resposta anterior como um parâmetro para obter a próxima parte da saída. Repita até que você não receba um `nextToken` valor na resposta.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Próxima parte da saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Exibir uma lista de IDs compilações \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário

- Para API obter detalhes, consulte [ListBuilds](#) na Referência de AWS CLI Comandos.

list-curated-environment-images

O código de exemplo a seguir mostra como usar `list-curated-environment-images`.

AWS CLI

Para obter uma lista de imagens do Docker gerenciadas por AWS CodeBuild ela, você pode usar em suas compilações.

O `list-curated-environment-images` exemplo a seguir lista as imagens do Docker gerenciadas pelo CodeBuild that podem ser usadas para compilações. :

```
aws codebuild list-curated-environment-images
```

Saída:

```
{
  "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 ...
          ]
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Para obter mais informações, consulte [Imagens do Docker fornecidas por CodeBuild](#) no Guia do AWS CodeBuild usuário

- Para API obter detalhes, consulte [ListCuratedEnvironmentImages](#) na Referência de AWS CLI Comandos.

list-projects

O código de exemplo a seguir mostra como usar `list-projects`.

AWS CLI

Para obter uma lista dos nomes dos projetos de AWS CodeBuild construção.

O `list-projects` exemplo a seguir obtém uma lista de projetos de CodeBuild construção classificados por nome em ordem crescente.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

A saída inclui um `nextToken` valor que indica que há mais saída disponível.

```
{
  "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"
  ]
}
```

Execute esse comando novamente e forneça o `nextToken` valor da resposta anterior como parâmetro para obter a próxima parte da saída. Repita até que você não receba um `nextToken` valor na resposta.

```
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"  
  ]  
}
```

Para obter mais informações, consulte [Exibir uma lista de nomes de projetos de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListProjects](#) na Referência de AWS CLI Comandos.

list-report-groups

O código de exemplo a seguir mostra como usar `list-report-groups`.

AWS CLI

Para obter uma lista do grupo de relatórios ARNs em AWS CodeBuild.

O `list-report-groups` exemplo a seguir recupera o grupo ARNs de relatórios da conta na região.

```
aws codebuild list-report-groups
```

Saída:

```
{  
  "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"  
  ]  
}
```

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListReportGroups](#) na Referência de AWS CLI Comandos.

list-reports-for-report-group

O código de exemplo a seguir mostra como usar `list-reports-for-report-group`.

AWS CLI

Para obter uma lista dos relatórios em um grupo de relatórios em AWS CodeBuild.

O `list-report-for-report-groups` exemplo a seguir recupera os relatórios no grupo de relatórios especificado para a conta na região.

```
aws codebuild list-reports-for-report-group \
  --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-
  group-name>
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListReportsForReportGroup](#) na Referência de AWS CLI Comandos.

list-reports

O código de exemplo a seguir mostra como usar `list-reports`.

AWS CLI

Para obter uma lista dos relatórios da conta corrente em AWS CodeBuild.

O `list-reports` exemplo a seguir recupera ARNs os relatórios da conta corrente.

```
aws codebuild list-reports
```

Saída:

```
{
  "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>"
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListReports](#) na Referência de AWS CLI Comandos.

`list-shared-projects`

O código de exemplo a seguir mostra como usar `list-shared-projects`.

AWS CLI

Para listar o projeto compartilhado em AWS CodeBuild.

O `list-shared-projects` exemplo a seguir lista os projetos CodeBuild compartilhados que estão disponíveis para a conta atual.

```
aws codebuild list-shared-projects
```

Saída:


```
{
  "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>"
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com projetos compartilhados](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListSharedProjects](#) na Referência de AWS CLI Comandos.

list-shared-report-groups

O código de exemplo a seguir mostra como usar `list-shared-report-groups`.

AWS CLI

Para obter uma lista do grupo de relatórios compartilhado ARNs em AWS CodeBuild.

O `list-shared-report-groups` exemplo a seguir recupera o grupo ARNs de relatórios da conta na região.

```
aws codebuild list-shared-report-groups
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListSharedReportGroups](#) na Referência de AWS CLI Comandos.

list-source-credentials

O código de exemplo a seguir mostra como usar `list-source-credentials`.

AWS CLI

Para ver uma lista de `sourceCredentialsObjects`

O `list-source-credentials` exemplo a seguir lista os tokens de uma AWS conta conectada a uma conta do Bitbucket e a uma GitHub conta. Cada `sourceCredentialsInfos` objeto na resposta contém informações de credenciais de origem conectada.

```
aws codebuild list-source-credentials
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Connect Source Providers with Access Tokens \(CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [ListSourceCredentials](#) na Referência de AWS CLI Comandos.

retry-build-batch

O código de exemplo a seguir mostra como usar `retry-build-batch`.

AWS CLI

Para tentar novamente a compilação de um lote com falha. AWS CodeBuild

O `retry-build-batch` exemplo a seguir reinicia a compilação em lote especificada.

```
aws codebuild retry-build-batch \  
  --id <project-name>:<batch-ID>
```

Saída:

```
{  
  "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"
    }
  }
]
}
}

```

Para obter mais informações, consulte [Batch builds in AWS CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [RetryBuildBatch](#) na Referência de AWS CLI Comandos.

retry-build

O código de exemplo a seguir mostra como usar `retry-build`.

AWS CLI

Para tentar novamente uma compilação com falha. AWS CodeBuild

O `retry-build` exemplo a seguir reinicia a compilação especificada.

```
aws codebuild retry-build \  
  --id <project-name>:<build-ID>
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Batch builds in AWS CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [RetryBuild](#) na Referência de AWS CLI Comandos.

start-build-batch

O código de exemplo a seguir mostra como usar `start-build-batch`.

AWS CLI

Para iniciar uma compilação em lote AWS CodeBuild.

O `start-build-batch` exemplo a seguir inicia uma construção em lote do projeto especificado.

```
aws codebuild start-build-batch \  
  --project-name <project-name>
```

Saída:

```
{  
  "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
}
}
```

Para obter mais informações, consulte [Batch builds in AWS CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [StartBuildBatch](#) na Referência de AWS CLI Comandos.

start-build

O código de exemplo a seguir mostra como usar `start-build`.

AWS CLI

Para começar a executar uma compilação de um projeto de AWS CodeBuild compilação.

O `start-build` exemplo a seguir inicia uma compilação para o CodeBuild projeto especificado. A compilação substitui a configuração do projeto pelo número de minutos em que a compilação pode ficar na fila antes que o tempo limite atinja o tempo limite e as configurações de artefato do projeto.

```
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}
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Executar uma compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [StartBuild](#) na Referência de AWS CLI Comandos.

stop-build-batch

O código de exemplo a seguir mostra como usar stop-build-batch.

AWS CLI

Para interromper a compilação de um lote em AWS CodeBuild andamento.

O stop-build-batch exemplo a seguir interrompe a criação em lote especificada.

```
aws codebuild stop-build-batch \  
  --id <project-name>:<batch-ID>
```

Saída:

```
{  
  "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"
        }
    }
]
}
}

```

Para obter mais informações, consulte [Batch builds in AWS CodeBuild](#) no Guia do AWS CodeBuild usuário.

- Para API obter detalhes, consulte [StopBuildBatch](#) na Referência de AWS CLI Comandos.

stop-build

O código de exemplo a seguir mostra como usar stop-build.

AWS CLI

Para interromper a compilação de um projeto de AWS CodeBuild compilação.

O stop-build exemplo a seguir interrompe a CodeBuild compilação especificada.


```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

Saída:

```
{
  "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"
}
```

```
}

```

Para obter mais informações, consulte [Parar uma compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [StopBuild](#) na Referência de AWS CLI Comandos.

update-project

O código de exemplo a seguir mostra como usar update-project.

AWS CLI

Para alterar as configurações de um projeto de AWS CodeBuild compilação.

O update-project exemplo a seguir altera as configurações do projeto de CodeBuild construção especificado chamado 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"
```

A saída exibe as configurações atualizadas.

```
{
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Alterar as configurações de um projeto de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário

- Para API obter detalhes, consulte [UpdateProject](#) na Referência de AWS CLI Comandos.

update-report-group

O código de exemplo a seguir mostra como usar update-report-group.

AWS CLI

Para atualizar um grupo de relatórios em AWS CodeBuild.

O `update-report-group` exemplo a seguir altera o tipo de exportação do grupo de relatórios para "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"
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Trabalhando com grupos de relatórios](#) no Guia AWS CodeBuild do usuário.

- Para API obter detalhes, consulte [UpdateReportGroup](#) na Referência de AWS CLI Comandos.

update-webhook

O código de exemplo a seguir mostra como usar `update-webhook`.

AWS CLI

Para atualizar o webhook de um projeto AWS CodeBuild

O `update-webhook` exemplo a seguir atualiza um webhook para o CodeBuild projeto especificado com dois grupos de filtros. O `--rotate-secret` parâmetro especifica que GitHub gire a chave secreta do projeto toda vez que uma alteração de código aciona uma compilação. O primeiro grupo de filtros especifica as solicitações pull que são criadas, atualizadas ou reabertas nas ramificações com nomes de referência do Git que correspondam à expressão regular `^refs/heads/master$` e referências head que correspondam a `^refs/heads/myBranch$`. O segundo grupo de filtros especifica solicitações push em ramificações com nomes de referência do Git que não correspondem à expressão regular `^refs/heads/myBranch$`

```
aws codebuild update-webhook \
  --project-name Project2 \
  --rotate-secret \
  --filter-groups "[[{"type":"EVENT","pattern":"PULL_REQUEST_CREATED,
PULL_REQUEST_UPDATED, PULL_REQUEST_REOPENED"}, {"type":"HEAD_REF","pattern
":"^refs/heads/myBranch$"},"excludeMatchedPattern":true}, {"type":"BASE_REF
","pattern":"^refs/heads/master$"},"excludeMatchedPattern":true}], [{"type":
"EVENT","pattern":"PUSH"}, {"type":"HEAD_REF","pattern":"^refs/heads/
myBranch$"},"excludeMatchedPattern":true}]]"
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Alterar as configurações de um projeto de compilação \(AWS CLI\)](#) no Guia AWS CodeBuild do usuário

- Para API obter detalhes, consulte [UpdateWebhook](#) na Referência de AWS CLI Comandos.

CodeCommit exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodeCommit.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-approval-rule-template-with-repository

O código de exemplo a seguir mostra como usar `associate-approval-rule-template-with-repository`.

AWS CLI

Para associar um modelo de regra de aprovação a um repositório

O `associate-approval-rule-template-with-repository` exemplo a seguir associa o modelo de regra de aprovação especificado a um repositório chamado `MyDemoRepo`

```
aws codecommit associate-approval-rule-template-with-repository \  
  --repository-name MyDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

Este comando não produz saída.

Para obter mais informações, consulte [Associar um modelo de regra de aprovação a um repositório](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [AssociateApprovalRuleTemplateWithRepository](#) na Referência de AWS CLI Comandos.

`batch-associate-approval-rule-template-with-repositories`

O código de exemplo a seguir mostra como usar `batch-associate-approval-rule-template-with-repositories`.

AWS CLI

Para associar um modelo de regra de aprovação a vários repositórios em uma única operação

O `batch-associate-approval-rule-template-with-repositories` exemplo a seguir associa o modelo de regra de aprovação especificado a repositórios chamados `MyDemoRepo` e `MyOtherDemoRepo`

Observação: os modelos de regras de aprovação são específicos para a AWS região em que foram criados. Eles só podem ser associados a repositórios nessa AWS região.

```
aws codecommit batch-associate-approval-rule-template-with-repositories \  
  --repository-names MyDemoRepo, MyOtherDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

Saída:

```
{
```

```

    "associatedRepositoryNames": [
      "MyDemoRepo",
      "MyOtherDemoRepo"
    ],
    "errors": []
  }

```

Para obter mais informações, consulte [Associar um modelo de regra de aprovação a um repositório](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [BatchAssociateApprovalRuleTemplateWithRepositories](#) na Referência de AWS CLI Comandos.

batch-describe-merge-conflicts

O código de exemplo a seguir mostra como usar `batch-describe-merge-conflicts`.

AWS CLI

Para obter informações sobre conflitos de mesclagem em todos os arquivos ou em um subconjunto de arquivos em uma mesclagem entre dois especificadores de confirmação

O `batch-describe-merge-conflicts` exemplo a seguir determina os conflitos de mesclagem para mesclar uma ramificação de origem nomeada `feature-randomizationfeature` com uma ramificação de destino nomeada `main` usando a `THREE_WAY_MERGE` estratégia em um repositório chamado `MyDemoRepo`

```

aws codecommit batch-describe-merge-conflicts \
  --source-commit-specifier feature-randomizationfeature \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE \
  --repository-name MyDemoRepo

```

Saída:

```

{
  "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"
}
```

Para obter mais informações, consulte [Resolver conflitos em uma pull request](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [BatchDescribeMergeConflicts](#) na Referência de AWS CLI Comandos.

batch-disassociate-approval-rule-template-from-repositories

O código de exemplo a seguir mostra como usar `batch-disassociate-approval-rule-template-from-repositories`.

AWS CLI

Para desassociar um modelo de regra de aprovação de vários repositórios em uma única operação

O `batch-disassociate-approval-rule-template-from-repositories` exemplo a seguir desassocia o modelo de regra de aprovação especificado dos repositórios chamados `e` e `MyDemoRepo` `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
```

Saída:

```
{
  "disassociatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Para obter mais informações, consulte [Desassociar um modelo de regra de aprovação](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [BatchDisassociateApprovalRuleTemplateFromRepositories](#) na Referência de AWS CLI Comandos.

batch-get-commits

O código de exemplo a seguir mostra como usar batch-get-commits.

AWS CLI

Para ver informações sobre vários commits

O batch-get-commits exemplo a seguir exibe detalhes sobre os commits especificados.

```
aws codecommit batch-get-commits \
  --repository-name MyDemoRepo \
  --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

Saída:

```
{
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Exibir detalhes da confirmação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [BatchGetCommits](#) na Referência de AWS CLI Comandos.

batch-get-repositories

O código de exemplo a seguir mostra como usar `batch-get-repositories`.

AWS CLI

Para ver detalhes sobre vários repositórios

Este exemplo mostra detalhes sobre vários AWS CodeCommit repositórios.

```
aws codecommit batch-get-repositories \
  --repository-names MyDemoRepo MyOtherDemoRepo
```

Saída:

```
{
```

```

    "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": []
  }

```

- Para API obter detalhes, consulte [BatchGetRepositories](#) na Referência de AWS CLI Comandos.

create-approval-rule-template

O código de exemplo a seguir mostra como usar create-approval-rule-template.

AWS CLI

Para criar um modelo de regra de aprovação

O `create-approval-rule-template` exemplo a seguir cria um modelo de regra de aprovação chamado `2-approver-rule-for-main`. The template requires two users who assume the role of `CodeCommitReview` para aprovar qualquer pull request antes que ela possa ser mesclada à main filial.

```
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/*"]}]}'
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Criar um modelo de regra de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [CreateApprovalRuleTemplate](#) na Referência de AWS CLI Comandos.

create-branch

O código de exemplo a seguir mostra como usar `create-branch`.

AWS CLI

Para criar uma ramificação

Este exemplo cria uma ramificação em um AWS CodeCommit repositório. Esse comando gerará saídas somente se houver erros.

Comando:

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
--commit-id 317f8570EXAMPLE
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [CreateBranch](#) na Referência de AWS CLI Comandos.

create-commit

O código de exemplo a seguir mostra como usar create-commit.

AWS CLI

Para criar um commit

O create-commit exemplo a seguir demonstra como criar uma confirmação inicial para um repositório que adiciona um readme.md arquivo a um repositório nomeado MyDemoRepo na ramificação. main

```
aws codecommit create-commit \
  --repository-name MyDemoRepo \
  --branch-name main \
  --put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

Saída:

```
{
  "filesAdded": [
    {
```

```

        "blobId": "5e1c309d-EXAMPLE",
        "absolutePath": "readme.md",
        "fileMode": "NORMAL"
    }
],
"commitId": "4df8b524-EXAMPLE",
"treeId": "55b57003-EXAMPLE",
"filesDeleted": [],
"filesUpdated": []
}

```

Para obter mais informações, consulte [Criar uma confirmação AWS CodeCommit no Guia do AWS CodeCommit usuário](#).

- Para API obter detalhes, consulte [CreateCommit](#) na Referência de AWS CLI Comandos.

create-pull-request-approval-rule

O código de exemplo a seguir mostra como usar `create-pull-request-approval-rule`.

AWS CLI

Para criar uma regra de aprovação para um pull request

O `create-pull-request-approval-rule` exemplo a seguir cria uma regra de aprovação com `Require two approved approvers` o nome da pull request especificada. A regra especifica que são necessárias duas aprovações de um grupo de aprovações. O pool inclui todos os usuários que acessam CodeCommit assumindo a função de `CodeCommitReview` na `123456789012` AWS conta. Também inclui um IAM usuário ou usuário federado chamado `Nikhil_Jayashankar` da mesma AWS conta.

```

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/*\"]}]}"

```

Saída:

```
{
```

```

"approvalRule": {
  "approvalRuleName": "Require two approved approvers",
  "lastModifiedDate": 1570752871.932,
  "ruleContentSha256": "7c44e6ebEXAMPLE",
  "creationDate": 1570752871.932,
  "approvalRuleId": "aac33506-EXAMPLE",
  "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":
[{\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}}]",
  "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
}
}

```

Para obter mais informações, consulte [Criar uma regra de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [CreatePullRequestApprovalRule](#) na Referência de AWS CLI Comandos.

create-pull-request

O código de exemplo a seguir mostra como usar create-pull-request.

AWS CLI

Para criar uma pull request

O create-pull-request exemplo a seguir cria uma pull request chamada “Analisador de dificuldade de pronúncia” com uma descrição de “Por favor, revise essas alterações até terça-feira” que tem como alvo a ramificação de origem 'jane-branch' e deve ser mesclada com a ramificação padrão 'main' em um repositório chamado ". 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

```

Saída:

```
{
```

```

"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
}
}

```

- Para API obter detalhes, consulte [CreatePullRequest](#) na Referência de AWS CLI Comandos.

create-repository

O código de exemplo a seguir mostra como usar `create-repository`.

AWS CLI

Para criar um repositório

Este exemplo cria um repositório e o associa à conta do AWS usuário.

Comando:

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-  
description "My demonstration repository"
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [CreateRepository](#) na Referência de AWS CLI Comandos.

create-unreferenced-merge-commit

O código de exemplo a seguir mostra como usar `create-unreferenced-merge-commit`.

AWS CLI

Para criar uma confirmação não referenciada que represente o resultado da fusão de dois especificadores de confirmação

O `create-unreferenced-merge-commit` exemplo a seguir cria um commit que representa os resultados de uma mesclagem entre uma ramificação de origem nomeada `bugfix-1234` com uma ramificação de destino nomeada `main` usando a MERGE estratégia `THREE_WAY __` em um repositório chamado. `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."
```

Saída:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Para obter mais informações, consulte [Resolver conflitos em uma pull request](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [CreateUnreferencedMergeCommit](#) na Referência de AWS CLI Comandos.

credential-helper

O código de exemplo a seguir mostra como usar `credential-helper`.

AWS CLI

Para configurar o auxiliar de credencial incluído no com AWS CLI AWS CodeCommit

O `credential-helper` utilitário não foi projetado para ser chamado diretamente do AWS CLI. Em vez disso, ele deve ser usado como um parâmetro com o `git config` comando para configurar seu computador local. Ele permite que o Git use HTTPS uma versão assinada criptograficamente de suas IAM credenciais de usuário ou EC2 função de instância da Amazon sempre que o Git precisar se autenticar para interagir com repositórios. AWS CodeCommit

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

Saída:

```
[credential]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

Para obter mais informações, consulte Configuração para AWS CodeCommit usar outros métodos no Guia do AWS CodeCommit usuário. Analise o conteúdo cuidadosamente e siga os procedimentos em um dos tópicos a seguir: Para HTTPS conexões no Linux, macOS ou Unix ou Para HTTPS conexões no Windows no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [CredentialHelper](#) na Referência de AWS CLI Comandos.

delete-approval-rule-template

O código de exemplo a seguir mostra como usar delete-approval-rule-template.

AWS CLI

Para excluir um modelo de regra de aprovação

O delete-approval-rule-template exemplo a seguir exclui o modelo de regra de aprovação especificado.

```
aws codecommit delete-approval-rule-template \
  --approval-rule-template-name 1-approver-for-all-pull-requests
```

Saída:

```
{
  "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

Para obter mais informações, consulte [Excluir um modelo de regra de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [DeleteApprovalRuleTemplate](#) na Referência de AWS CLI Comandos.

delete-branch

O código de exemplo a seguir mostra como usar `delete-branch`.

AWS CLI

Para excluir uma ramificação

Este exemplo mostra como excluir uma ramificação em um AWS CodeCommit repositório.

Comando:

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Saída:

```
{
  "branch": {
    "commitId": "317f8570EXAMPLE",
    "branchName": "MyNewBranch"
  }
}
```

- Para API obter detalhes, consulte [DeleteBranch](#) na Referência de AWS CLI Comandos.

delete-comment-content

O código de exemplo a seguir mostra como usar `delete-comment-content`.

AWS CLI

Para excluir o conteúdo de um comentário

Você só pode excluir o conteúdo de um comentário se você criou o comentário. Este exemplo demonstra como excluir o conteúdo de um comentário com o ID gerado pelo sistema de. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit delete-comment-content \
  --comment-id ff30b348EXAMPLEb9aa670f
```

Saída:


```
{
  "comment": {
    "creationDate": 1508369768.142,
    "deleted": true,
    "lastModifiedDate": 1508369842.278,
    "clientRequestToken": "123Example",
    "commentId": "ff30b348EXAMPLEeb9aa670f",
    "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
    "callerReactions": [],
    "reactionCounts":
    {
      "CLAP" : 1
    }
  }
}
```

- Para API obter detalhes, consulte [DeleteCommentContent](#) na Referência de AWS CLI Comandos.

delete-file

O código de exemplo a seguir mostra como usar delete-file.

AWS CLI

Para excluir um arquivo

O delete-file exemplo a seguir demonstra como excluir um arquivo chamado README.md de uma ramificação nomeada main com uma ID de confirmação mais recente de c5709475EXAMPLE em um repositório chamado. MyDemoRepo

```
aws codecommit delete-file \
  --repository-name MyDemoRepo \
  --branch-name main \
  --file-path README.md \
  --parent-commit-id c5709475EXAMPLE
```

Saída:

```
{
  "blobId": "559b44fEXAMPLE",
```

```
"commitId":"353cf655EXAMPLE",
"filePath":"README.md",
"treeId":"6bc824cEXAMPLE"
}
```

Para obter mais informações, consulte [Editar ou excluir um arquivo AWS CodeCommit no Guia de AWS CodeCommit API](#) referência.

- Para API obter detalhes, consulte [DeleteFile](#) na Referência de AWS CLI Comandos.

delete-pull-request-approval-rule

O código de exemplo a seguir mostra como usar `delete-pull-request-approval-rule`.

AWS CLI

Para excluir uma regra de aprovação para um pull request

O `delete-pull-request-approval-rule` exemplo a seguir exclui a regra de aprovação nomeada `My Approval Rule` para a pull request especificada.

```
aws codecommit delete-pull-request-approval-rule \
  --approval-rule-name "My Approval Rule" \
  --pull-request-id 15
```

Saída:

```
{
  "approvalRuleId": "077d8e8a8-EXAMPLE"
}
```

Para obter mais informações, consulte [Editar ou excluir uma regra de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [DeletePullRequestApprovalRule](#) na Referência de AWS CLI Comandos.

delete-repository

O código de exemplo a seguir mostra como usar `delete-repository`.

AWS CLI

Para excluir um repositório

Este exemplo mostra como excluir um AWS CodeCommit repositório.

Comando:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

Saída:

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

- Para API obter detalhes, consulte [DeleteRepository](#) na Referência de AWS CLI Comandos.

describe-merge-conflicts

O código de exemplo a seguir mostra como usar describe-merge-conflicts.

AWS CLI

Para obter informações detalhadas sobre conflitos de mesclagem

O describe-merge-conflicts exemplo a seguir determina os conflitos de mesclagem de um arquivo nomeado `readme.md` na ramificação de origem e na ramificação de destino especificadas usando a MERGE estratégia `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
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Resolver conflitos em uma pull request](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [DescribeMergeConflicts](#) na Referência de AWS CLI Comandos.

describe-pull-request-events

O código de exemplo a seguir mostra como usar `describe-pull-request-events`.

AWS CLI

Para visualizar eventos em uma pull request

O `describe-pull-request-events` exemplo a seguir recupera os eventos de uma pull request com o ID '8'.

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Saída:

```
{
  "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"
      }
    ]
  }
```

- Para API obter detalhes, consulte [DescribePullRequestEvents](#) na Referência de AWS CLI Comandos.

disassociate-approval-rule-template-from-repository

O código de exemplo a seguir mostra como usar `disassociate-approval-rule-template-from-repository`.

AWS CLI

Para desassociar um modelo de regra de aprovação de um repositório

O `disassociate-approval-rule-template-from-repository` exemplo a seguir desassocia o modelo de regra de aprovação especificado de um repositório chamado

`MyDemoRepo`

```
aws codecommit disassociate-approval-rule-template-from-repository \
  --repository-name MyDemoRepo \
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Este comando não produz saída.

Para obter mais informações, consulte [Desassociar um modelo de regra de aprovação](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [DisassociateApprovalRuleTemplateFromRepository](#) na Referência de AWS CLI Comandos.

evaluate-pull-request-approval-rules

O código de exemplo a seguir mostra como usar `evaluate-pull-request-approval-rules`.

AWS CLI

Para avaliar se uma pull request tem todas as suas regras de aprovação satisfeitas

O `evaluate-pull-request-approval-rules` exemplo a seguir avalia o estado das regras de aprovação na pull request especificada. Neste exemplo, uma regra de aprovação não foi satisfeita para a pull request, então a saída do comando mostra um `approved` valor `false`.

```
aws codecommit evaluate-pull-request-approval-rules \  
  --pull-request-id 27 \  
  --revision-id 9f29d167EXAMPLE
```

Saída:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ],  
    "overridden": false,  
    "approvalRulesSatisfied": []  
  }  
}
```

Para obter mais informações, consulte [Mesclar uma pull request](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [EvaluatePullRequestApprovalRules](#) na Referência de AWS CLI Comandos.

get-approval-rule-template

O código de exemplo a seguir mostra como usar `get-approval-rule-template`.

AWS CLI

Para obter o conteúdo de um modelo de regra de aprovação

O `get-approval-rule-template` exemplo a seguir obtém o conteúdo de um modelo de regra de aprovação chamado `1-approver-rule-for-all-pull-requests`.

```
aws codecommit get-approval-rule-template \  
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Saída:

```
{
  "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."
  }
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [GetApprovalRuleTemplate](#) na Referência de AWS CLI Comandos.

get-blob

O código de exemplo a seguir mostra como usar `get-blob`.

AWS CLI

Para visualizar informações sobre um objeto blob do Git

O `get-blob` exemplo a seguir recupera informações sobre um blob Git com o ID de EXAMPLE '2eb4af3b' em um repositório chamado ' '. AWS CodeCommit MyDemoRepo

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```

Saída:

```
{
  "content": "QSBCaW5hcnkgTGFyToEXAMPLE="
}
```


- Para API obter detalhes, consulte [GetBlob](#) na Referência de AWS CLI Comandos.

get-branch

O código de exemplo a seguir mostra como usar `get-branch`.

AWS CLI

Para obter informações sobre uma filial

Este exemplo obtém informações sobre uma ramificação em um AWS CodeCommit repositório.

Comando:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

Saída:

```
{
  "BranchInfo": {
    "commitID": "317f8570EXAMPLE",
    "branchName": "MyNewBranch"
  }
}
```

- Para API obter detalhes, consulte [GetBranch](#) na Referência de AWS CLI Comandos.

get-comment-reactions

O código de exemplo a seguir mostra como usar `get-comment-reactions`.

AWS CLI

Para ver as reações de emoji a um comentário

O `get-comment-reactions` exemplo a seguir lista todas as reações de emoji a um comentário com o ID `deabcd1234EXAMPLEb5678efgh`. Se a fonte do seu shell suportar a exibição do Emoji versão 1.0, a saída `emoji` do emoji será exibida.

```
aws codecommit get-comment-reactions \  
  --comment-id abcd1234EXAMPLEb5678efgh
```

Saída:

```
{
  "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"
        ]
      }
    ]
  }
}
```

Para obter mais informações, consulte [Comentar sobre uma confirmação AWS CodeCommit no Guia do AWS CodeCommit usuário](#).

- Para API obter detalhes, consulte [GetCommentReactions](#) na Referência de AWS CLI Comandos.

get-comment

O código de exemplo a seguir mostra como usar `get-comment`.

AWS CLI

Para ver os detalhes de um comentário

Este exemplo demonstra como visualizar os detalhes de um comentário com o ID de comentário gerado pelo sistema de. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit get-comment \  
  --comment-id ff30b348EXAMPLEb9aa670f
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetComment](#) na Referência de AWS CLI Comandos.

get-comments-for-compared-commit

O código de exemplo a seguir mostra como usar `get-comments-for-compared-commit`.

AWS CLI

Para ver comentários em um commit

Este exemplo demonstra como visualizar os comentários feitos na comparação entre dois commits em um repositório chamado. MyDemoRepo

```
aws codecommit get-comments-for-compared-commit \
  --repository-name MyDemoRepo \
  --before-commit-ID 6e147360EXAMPLE \
  --after-commit-id 317f8570EXAMPLE
```

Saída:

```
{
  "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"
}

```

- Para API obter detalhes, consulte [GetCommentsForComparedCommit](#) na Referência de AWS CLI Comandos.

get-comments-for-pull-request

O código de exemplo a seguir mostra como usar `get-comments-for-pull-request`.

AWS CLI

Para ver os comentários de um pull request

Este exemplo demonstra como visualizar os comentários de uma pull request em um repositório chamado. MyDemoRepo

```
aws codecommit get-comments-for-pull-request \
```

```

--repository-name MyDemoRepo \
--before-commit-ID 317f8570EXAMPLE \
--after-commit-id 5d036259EXAMPLE

```

Saída:

```

{
  "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"
}

```

- Para API obter detalhes, consulte [GetCommentsForPullRequest](#) na Referência de AWS CLI Comandos.

get-commit

O código de exemplo a seguir mostra como usar `get-commit`.

AWS CLI

Para visualizar informações sobre um commit em um repositório

Este exemplo mostra detalhes sobre um commit com o ID gerado pelo sistema de '7e9fd3091thisisanexamplethisisanexample1' em um repositório chamado ". AWS CodeCommit MyDemoRepo

Comando:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id 7e9fd3091thisisanexamplethisisanexample1
```

Saída:

```

{
  "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"
}
}

```

- Para API obter detalhes, consulte [GetCommit](#) na Referência de AWS CLI Comandos.

get-differences

O código de exemplo a seguir mostra como usar `get-differences`.

AWS CLI

Para obter informações sobre as diferenças de um especificador de confirmação em um repositório

Este exemplo mostra informações de metadados de visualização sobre alterações entre dois especificadores de confirmação (ramificação HEAD, tag ou outras referências totalmente qualificadas, como confirmaçãoIDs) em uma pasta renomeada no AWS CodeCommit repositório chamado. MyDemoRepo O exemplo inclui várias opções que não são obrigatórias, incluindo `--before-commit-specifier`, `--before-path` e `--after-path`, para ilustrar melhor como você pode usar essas opções para limitar os resultados. A resposta inclui permissões do modo de arquivo.

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

```

Saída:


```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
        "mode": "100644"
      },
      "changeType": "M",
      "beforeBlob": {
        "path": "blob.txt",
        "blobId": "bf7fcf281thisisanexamplethisisanexample1",
        "mode": "100644"
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [GetDifferences](#) na Referência de AWS CLI Comandos.

get-file

O código de exemplo a seguir mostra como usar `get-file`.

AWS CLI

Para obter o conteúdo codificado em base 64 de um arquivo em um repositório AWS CodeCommit

O `get-file` exemplo a seguir demonstra como obter o conteúdo codificado em base 64 de um arquivo chamado `README.md` de uma ramificação nomeada `main` em um repositório chamado `MyDemoRepo`

```
aws codecommit get-file \
  --repository-name MyDemoRepo \
  --commit-specifier main \
  --file-path README.md
```

Saída:

```
{
```

```
"blobId":"559b44fEXAMPLE",
"commitId":"c5709475EXAMPLE",
"fileContent":"IyBQaHVzEXAMPLE",
"filePath":"README.md",
"fileMode":"NORMAL",
"fileSize":1563
}
```

Para obter mais informações, consulte [GetFile](#) no Guia AWS CodeCommit API de referência.

- Para API obter detalhes, consulte [GetFile](#) na Referência de AWS CLI Comandos.

get-folder

O código de exemplo a seguir mostra como usar `get-folder`.

AWS CLI

Para obter o conteúdo de uma pasta em um AWS CodeCommit repositório

O `get-folder` exemplo a seguir demonstra como obter o conteúdo de uma pasta de nível superior de um repositório chamado. `MyDemoRepo`

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [GetFolder](#) o Guia AWS CodeCommit API de referência.

- Para API obter detalhes, consulte [GetFolder](#) na Referência de AWS CLI Comandos.

get-merge-commit

O código de exemplo a seguir mostra como usar `get-merge-commit`.

AWS CLI

Para obter informações detalhadas sobre uma confirmação de mesclagem

O `get-merge-commit` exemplo a seguir exibe detalhes sobre uma confirmação de mesclagem para a ramificação de origem nomeada `bugfix-bug1234` com uma ramificação de destino nomeada `main` usando a MERGE estratégia `THREE_WAY_MERGE` em um repositório chamado `MyDemoRepo`

```
aws codecommit get-merge-commit \  
  --source-commit-specifier bugfix-bug1234 \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE \  
  --repository-name MyDemoRepo
```

Saída:

```
{  
  "sourceCommitId": "c5709475EXAMPLE",  
  "destinationCommitId": "317f8570EXAMPLE",  
  "baseCommitId": "fb12a539EXAMPLE",  
  "mergeCommitId": "ffc4d608eEXAMPLE"  
}
```

Para obter mais informações, consulte [Exibir detalhes da confirmação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [GetMergeCommit](#) na Referência de AWS CLI Comandos.

get-merge-conflicts

O código de exemplo a seguir mostra como usar `get-merge-conflicts`.

AWS CLI

Para ver se há algum conflito de mesclagem em uma pull request

O `get-merge-conflicts` exemplo a seguir mostra se há algum conflito de mesclagem entre a ponta de uma ramificação de origem chamada `feature-randomizationfeature` e uma ramificação de destino chamada `'main'` em um repositório chamado `MyDemoRepo`

```
aws codecommit get-merge-conflicts \  
  --repository-name MyDemoRepo \  
  --source-commit-specifier feature-randomizationfeature \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

```

    }
  }
]
}

```

- Para API obter detalhes, consulte [GetMergeConflicts](#) na Referência de AWS CLI Comandos.

get-merge-options

O código de exemplo a seguir mostra como usar `get-merge-options`.

AWS CLI

Para obter informações sobre as opções de mesclagem disponíveis para mesclar duas ramificações especificadas

O `get-merge-options` exemplo a seguir determina as opções de mesclagem disponíveis para mesclar uma ramificação de origem nomeada `bugfix-bug1234` com uma ramificação de destino nomeada `main` em um repositório chamado `MyDemoRepo`

```

aws codecommit get-merge-options \
  --source-commit-specifier bugfix-bug1234 \
  --destination-commit-specifier main \
  --repository-name MyDemoRepo

```

Saída:

```

{
  "mergeOptions": [
    "FAST_FORWARD_MERGE",
    "SQUASH_MERGE",
    "THREE_WAY_MERGE"
  ],
  "sourceCommitId": "18059494EXAMPLE",
  "destinationCommitId": "ffd3311dEXAMPLE",
  "baseCommitId": "ffd3311dEXAMPLE"
}

```

Para obter mais informações, consulte [Resolver conflitos em uma pull request](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [GetMergeOptions](#) na Referência de AWS CLI Comandos.

get-pull-request-approval-states

O código de exemplo a seguir mostra como usar `get-pull-request-approval-states`.

AWS CLI

Para ver as aprovações em uma pull request

O `get-pull-request-approval-states` exemplo a seguir retorna aprovações para a pull request especificada.

```
aws codecommit get-pull-request-approval-states \  
  --pull-request-id 8 \  
  --revision-id 9f29d167EXAMPLE
```

Saída:

```
{  
  "approvals": [  
    {  
      "userArn": "arn:aws:iam::123456789012:user/Mary_Major",  
      "approvalState": "APPROVE"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Exibir pull requests](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [GetPullRequestApprovalStates](#) na Referência de AWS CLI Comandos.

get-pull-request-override-state

O código de exemplo a seguir mostra como usar `get-pull-request-override-state`.

AWS CLI

Para obter informações sobre o status de substituição de uma pull request

O `get-pull-request-override-state` exemplo a seguir retorna o estado de substituição da pull request especificada. Neste exemplo, as regras de aprovação da pull request foram substituídas por uma usuária chamada Mary Major, então a saída retorna um valor de `true` :

```
aws codecommit get-pull-request-override-state \  
  --pull-request-id 34 \  
  --revision-id 9f29d167EXAMPLE
```

Saída:

```
{  
  "overridden": true,  
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"  
}
```

Para obter mais informações, consulte [Substituir regras de aprovação em uma pull request](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [GetPullRequestOverrideState](#) na Referência de AWS CLI Comandos.

get-pull-request

O código de exemplo a seguir mostra como usar `get-pull-request`.

AWS CLI

Para ver os detalhes de uma pull request

Este exemplo demonstra como visualizar informações sobre uma pull request com o ID de 27.

```
aws codecommit get-pull-request \  
  --pull-request-id 27
```

Saída:

```
{  
  "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"
}
}

```

- Para API obter detalhes, consulte [GetPullRequest](#) na Referência de AWS CLI Comandos.

get-repository-triggers

O código de exemplo a seguir mostra como usar `get-repository-triggers`.

AWS CLI

Para obter informações sobre gatilhos em um repositório

Este exemplo mostra detalhes sobre acionadores configurados para um AWS CodeCommit repositório chamado. MyDemoRepo

```
aws codecommit get-repository-triggers \  
  --repository-name MyDemoRepo
```

Saída:

```
{  
  "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"  
      ]  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [GetRepositoryTriggers](#) na Referência de AWS CLI Comandos.

get-repository

O código de exemplo a seguir mostra como usar `get-repository`.

AWS CLI

Para obter informações sobre um repositório

Este exemplo mostra detalhes sobre um AWS CodeCommit repositório.

```
aws codecommit get-repository \  
  --repository-name MyDemoRepo
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [GetRepository](#) na Referência de AWS CLI Comandos.

list-approval-rule-templates

O código de exemplo a seguir mostra como usar `list-approval-rule-templates`.

AWS CLI

Para listar todos os modelos de regras de aprovação em uma AWS região

O `list-approval-rule-templates` exemplo a seguir lista todos os modelos de regras de aprovação na região especificada. Se nenhuma AWS região for especificada como parâmetro, o comando retornará modelos de regras de aprovação para a região especificada no AWS CLI perfil usado para executar o comando.

```
aws codecommit list-approval-rule-templates \  
  --region us-east-2
```

Saída:

```
{  
  "approvalRuleTemplateName": [  
    "2-approver-rule-for-main",  
    "1-approver-rule-for-all-pull-requests"  
  ]  
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [ListApprovalRuleTemplates](#) na Referência de AWS CLI Comandos.

list-associated-approval-rule-templates-for-repository

O código de exemplo a seguir mostra como usar `list-associated-approval-rule-templates-for-repository`.

AWS CLI

Para listar todos os modelos associados a um repositório

O `list-associated-approval-rule-templates-for-repository` exemplo a seguir lista todos os modelos de regras de aprovação associados a um repositório chamado `MyDemoRepo`.

```
aws codecommit list-associated-approval-rule-templates-for-repository \  
  --repository-name MyDemoRepo
```

Saída:

```
{  
  "approvalRuleTemplateName": [  
    "2-approver-rule-for-main",  
    "1-approver-rule-for-all-pull-requests"  
  ]  
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [ListAssociatedApprovalRuleTemplatesForRepository](#) na Referência de AWS CLI Comandos.

list-branches

O código de exemplo a seguir mostra como usar `list-branches`.

AWS CLI

Para ver uma lista de nomes de filiais

Este exemplo lista todos os nomes de ramificações em um AWS CodeCommit repositório.

```
aws codecommit list-branches \  
  --repository-name MyDemoRepo
```

Saída:

```
{  
  "branches": [  
    "MyNewBranch",  
    "main"  
  ]  
}
```

- Para API obter detalhes, consulte [ListBranches](#) na Referência de AWS CLI Comandos.

list-pull-requests

O código de exemplo a seguir mostra como usar `list-pull-requests`.

AWS CLI

Para ver uma lista de pull requests em um repositório

Este exemplo demonstra como listar pull requests criadas por um IAM usuário com o ARN 'arn:aws:iam: :111111111111:user/li_juan' e o status de " em um repositório chamado ": 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
```

Saída:

```
{
  "nextToken": "",
  "pullRequestIds": ["2", "12", "16", "22", "23", "35", "30", "39", "47"]
}
```

- Para API obter detalhes, consulte [ListPullRequests](#) na Referência de AWS CLI Comandos.

list-repositories-for-approval-rule-template

O código de exemplo a seguir mostra como usar `list-repositories-for-approval-rule-template`.

AWS CLI

Para listar todos os repositórios associados a um modelo

O `list-repositories-for-approval-rule-template` exemplo a seguir lista todos os repositórios associados ao modelo de regra de aprovação especificado.

```
aws codecommit list-repositories-for-approval-rule-template \
  --approval-rule-template-name 2-approver-rule-for-main
```

Saída:

```
{
  "repositoryNames": [
    "MyDemoRepo",
    "MyClonedRepo"
  ]
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [ListRepositoriesForApprovalRuleTemplate](#) na Referência de AWS CLI Comandos.

list-repositories

O código de exemplo a seguir mostra como usar `list-repositories`.

AWS CLI

Para ver uma lista de repositórios

Este exemplo lista todos os AWS CodeCommit repositórios associados à AWS conta do usuário.

Comando:

```
aws codecommit list-repositories
```

Saída:

```
{
  "repositories": [
    {
      "repositoryName": "MyDemoRepo"
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
    },
    {
      "repositoryName": "MyOtherDemoRepo"
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListRepositories](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para visualizar as AWS tags de um repositório

O `list-tags-for-resource` exemplo a seguir lista as chaves e os valores das tags para o repositório especificado.

```
aws codecommit list-tags-for-resource \  
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

Saída:

```
{  
  "tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

Para obter mais informações, consulte [Exibir tags de um repositório](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

merge-branches-by-fast-forward

O código de exemplo a seguir mostra como usar `merge-branches-by-fast-forward`.

AWS CLI

Para mesclar duas ramificações usando a estratégia de mesclagem rápida

O `merge-branches-by-fast-forward` exemplo a seguir mescla a ramificação de origem especificada com a ramificação de destino especificada em um repositório chamado `MyDemoRepo`

```
aws codecommit merge-branches-by-fast-forward \  
  --source-commit-specifier bugfix-bug1234 \  
  --destination-commit-specifier bugfix-bug1233 \  
  --repository-name MyDemoRepo
```

Saída:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```


Para obter mais informações, consulte [Comparar e mesclar ramificações](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [MergeBranchesByFastForward](#) na Referência de AWS CLI Comandos.

merge-branches-by-squash

O código de exemplo a seguir mostra como usar merge-branches-by-squash.

AWS CLI

Para mesclar duas ramificações usando a estratégia de fusão de squash

O merge-branches-by-squash exemplo a seguir mescla a ramificação de origem especificada com a ramificação de destino especificada em um repositório chamado 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
```

Saída:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Para obter mais informações, consulte [Comparar e mesclar ramificações](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [MergeBranchesBySquash](#) na Referência de AWS CLI Comandos.

merge-branches-by-three-way

O código de exemplo a seguir mostra como usar merge-branches-by-three-way.

AWS CLI

Para mesclar duas ramificações usando a estratégia de mesclagem de três vias

O `merge-branches-by-three-way` exemplo a seguir mescla a ramificação de origem especificada com a ramificação de destino especificada em um repositório chamado.

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
```

Saída:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Para obter mais informações, consulte [Comparar e mesclar ramificações](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [MergeBranchesByThreeWay](#) na Referência de AWS CLI Comandos.

`merge-pull-request-by-fast-forward`

O código de exemplo a seguir mostra como usar `merge-pull-request-by-fast-forward`.

AWS CLI

Para mesclar e fechar uma pull request

Este exemplo demonstra como mesclar e fechar uma pull request com o ID '47' e um ID de confirmação de origem '99132ab0' EXAMPLE em um repositório chamado. MyDemoRepo

```
aws codecommit merge-pull-request-by-fast-forward \  
  --pull-request-id 47 --repository-name MyDemoRepo
```

```
--pull-request-id 47 \
--source-commit-id 99132ab0EXAMPLE \
--repository-name MyDemoRepo
```

Saída:

```
{
  "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"
  }
}
```

```
}
}
```

Para obter mais informações, consulte [Mesclar uma pull request](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [MergePullRequestByFastForward](#) na Referência de AWS CLI Comandos.

merge-pull-request-by-squash

O código de exemplo a seguir mostra como usar `merge-pull-request-by-squash`.

AWS CLI

Para mesclar uma pull request usando a estratégia de mesclagem de squash

O `merge-pull-request-by-squash` exemplo a seguir mescla e fecha a pull request especificada usando a estratégia de resolução de conflitos de `ACCEPT_SOURCE` em um repositório chamado. `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"
```

Saída:

```
{
  "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"
}
}

```

Para obter mais informações, consulte [Mesclar uma pull request](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [MergePullRequestBySquash](#) na Referência de AWS CLI Comandos.

merge-pull-request-by-three-way

O código de exemplo a seguir mostra como usar `merge-pull-request-by-three-way`.

AWS CLI

Para mesclar uma pull request usando a estratégia de mesclagem de três vias

O `merge-pull-request-by-three-way` exemplo a seguir mescla e fecha a pull request especificada usando as opções padrão para detalhes do conflito e estratégia de resolução de conflitos em um repositório chamado. `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"
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Mesclar uma pull request](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [MergePullRequestByThreeWay](#) na Referência de AWS CLI Comandos.

override-pull-request-approval-rules

O código de exemplo a seguir mostra como usar `override-pull-request-approval-rules`.

AWS CLI

Para substituir os requisitos da regra de aprovação em um pull request

O `override-pull-request-approval-rules` exemplo a seguir substitui as regras de aprovação na pull request especificada. Em vez disso, para revogar uma substituição, defina o valor do `--override-status` parâmetro como `REVOKE`

```
aws codecommit override-pull-request-approval-rules \
```

```
--pull-request-id 34 \  
--revision-id 927df8d8EXAMPLE \  
--override-status OVERRIDE
```

Este comando não produz saída.

Para obter mais informações, consulte [Substituir regras de aprovação em uma pull request](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [OverridePullRequestApprovalRules](#) na Referência de AWS CLI Comandos.

post-comment-for-compared-commit

O código de exemplo a seguir mostra como usar `post-comment-for-compared-commit`.

AWS CLI

Para criar um comentário em um commit

Este exemplo demonstra como adicionar o comentário "Can you add a test case for this?" sobre a alteração no `cl_sample.js` arquivo na comparação entre dois commits em um repositório chamado. MyDemoRepo

```
aws codecommit post-comment-for-compared-commit \  
--repository-name MyDemoRepo \  
--before-commit-id 317f8570EXAMPLE \  
--after-commit-id 5d036259EXAMPLE \  
--client-request-token 123Example \  
--content "Can you add a test case for this?" \  
--location filePath=cl_sample.js,filePosition=1232,relativeFileVersion=AFTER
```

Saída:

```
{  
  "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"
}

```

- Para API obter detalhes, consulte [PostCommentForComparedCommit](#) na Referência de AWS CLI Comandos.

post-comment-for-pull-request

O código de exemplo a seguir mostra como usar `post-comment-for-pull-request`.

AWS CLI

Para adicionar um comentário a uma pull request

O `post-comment-for-pull-request` exemplo a seguir adiciona o comentário “Eles não parecem ser usados em nenhum lugar. Podemos removê-los?” sobre a alteração no `ahs_count.py` arquivo em uma pull request com o ID de 47 em um repositório chamado `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

```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [PostCommentForPullRequest](#) na Referência de AWS CLI Comandos.

post-comment-reply

O código de exemplo a seguir mostra como usar `post-comment-reply`.

AWS CLI

Para responder a um comentário em um commit ou em uma pull request

Este exemplo demonstra como adicionar a resposta "Good catch. I'll remove them." ao comentário com o ID gerado pelo sistema de. `abcd1234EXAMPLEb5678efgh`

```
aws codecommit post-comment-reply \  
  --in-reply-to abcd1234EXAMPLEeb5678efgh \  
  --content "Good catch. I'll remove them." \  
  --client-request-token 123Example
```

Saída:

```
{  
  "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": "abcd1234EXAMPLEeb5678efgh",  
    "lastModifiedDate": 150836912.221,  
    "callerReactions": [],  
    "reactionCounts": []  
  }  
}
```

- Para API obter detalhes, consulte [PostCommentReply](#) na Referência de AWS CLI Comandos.

put-comment-reaction

O código de exemplo a seguir mostra como usar `put-comment-reaction`.

AWS CLI

Para responder a um comentário em um commit com um emoji

O `put-comment-reaction` exemplo a seguir responde a um comentário com o ID de `abcd1234EXAMPLEeb5678efgh` com um valor de reação de emoji de `:thumbsup:`

```
aws codecommit put-comment-reaction \  
  --comment-id abcd1234EXAMPLEeb5678efgh \  
  --reaction-value :thumbsup:
```

Este comando não produz saída.

Para obter mais informações, consulte [Comentar sobre uma confirmação AWS CodeCommit no Guia do AWS CodeCommit usuário](#).

- Para API obter detalhes, consulte [PutCommentReaction](#) na Referência de AWS CLI Comandos.

put-file

O código de exemplo a seguir mostra como usar `put-file`.

AWS CLI

Para adicionar um arquivo a um repositório

O `put-file` exemplo a seguir adiciona um arquivo chamado 'ExampleSolution.py' a um repositório chamado " a uma ramificação chamada MyDemoRepo 'feature-randomizationfeature' cujo commit mais recente tem um ID de '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."
```

Saída:

```
{  
  "blobId": "2eb4af3bEXAMPLE",  
  "commitId": "317f8570EXAMPLE",  
  "treeId": "347a3408EXAMPLE"  
}
```

- Para API obter detalhes, consulte [PutFile](#) na Referência de AWS CLI Comandos.

put-repository-triggers

O código de exemplo a seguir mostra como usar `put-repository-triggers`.

AWS CLI

Para adicionar ou atualizar um gatilho em um repositório

Este exemplo demonstra como atualizar acionadores chamados " e 'MyFirstTriggerMySecondTrigger' usando um JSON arquivo já criado (aqui chamado MyTriggers .json) que contém a estrutura de todos os acionadores de um repositório chamado MyDemoRepo Para saber como obter os JSON acionadores existentes, consulte o get-repository-triggers comando.

```
aws codecommit put-repository-triggers \  
  --repository-name MyDemoRepo file://MyTriggers.json
```

Conteúdo de 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"  
      ]  
    }  
  ]  
}
```

```
}
```

Saída:

```
{  
  "configurationId": "6fa51cd8-35c1-EXAMPLE"  
}
```

- Para API obter detalhes, consulte [PutRepositoryTriggers](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar AWS tags a um repositório existente

O `tag-resource` exemplo a seguir marca o repositório especificado com duas tags.

```
aws codecommit tag-resource \  
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \  
  --tags Status=Secret,Team=Saarvi
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma tag a um repositório](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

test-repository-triggers

O código de exemplo a seguir mostra como usar `test-repository-triggers`.

AWS CLI

Para testar gatilhos em um repositório

Este exemplo demonstra como testar um gatilho chamado 'MyFirstTrigger' em um AWS CodeCommit repositório chamado. MyDemoRepo Neste exemplo, eventos no repositório acionam notificações de um tópico do 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
```

Saída:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- Para API obter detalhes, consulte [TestRepositoryTriggers](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover AWS tags de um repositório

O untag-resource exemplo a seguir remove a tag com a chave especificada do repositório chamado MyDemoRepo.

```
aws codecommit untag-resource \
--resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
--tag-keys Status
```

Este comando não produz saída.

Para obter mais informações, consulte [Remover uma tag de um repositório](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-approval-rule-template-content

O código de exemplo a seguir mostra como usar `update-approval-rule-template-content`.

AWS CLI

Para atualizar o conteúdo de um modelo de regra de aprovação

O `update-approval-rule-template-content` exemplo a seguir altera o conteúdo do modelo de regra de aprovação especificado para redefinir o grupo de aprovação para usuários que assumem a função de `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/*\"]}]}"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [UpdateApprovalRuleTemplateContent](#) na Referência de AWS CLI Comandos.

update-approval-rule-template-description

O código de exemplo a seguir mostra como usar `update-approval-rule-template-description`.

AWS CLI

Para atualizar a descrição de um modelo de regra de aprovação

O `update-approval-rule-template-description` exemplo a seguir altera a descrição do modelo de regra de aprovação especificado para `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"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [UpdateApprovalRuleTemplateDescription](#) na Referência de AWS CLI Comandos.

update-approval-rule-template-name

O código de exemplo a seguir mostra como usar `update-approval-rule-template-name`.

AWS CLI

Para atualizar o nome de um modelo de regra de aprovação

O `update-approval-rule-template-name` exemplo a seguir altera o nome de um modelo de regra de aprovação de `1-approver-rule` para `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
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Gerenciar modelos de regras de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [UpdateApprovalRuleTemplateName](#) na Referência de AWS CLI Comandos.

update-comment

O código de exemplo a seguir mostra como usar `update-comment`.

AWS CLI

Para atualizar um comentário em um commit

Este exemplo demonstra como adicionar o conteúdo "Fixed as requested. I'll update the pull request." a um comentário com um ID de 442b498bEXAMPLE5756813.

```
aws codecommit update-comment \  
  --comment-id 442b498bEXAMPLE5756813 \  
  --content "Fixed as requested. I'll update the pull request."
```

Saída:

```
{  
  "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  
      }  
  }  
}
```

- Para API obter detalhes, consulte [UpdateComment](#) na Referência de AWS CLI Comandos.

update-default-branch

O código de exemplo a seguir mostra como usar `update-default-branch`.

AWS CLI

Para alterar a ramificação padrão de um repositório

Este exemplo altera a ramificação padrão de um AWS CodeCommit repositório. Esse comando gerará saídas somente se houver erros.

Comando:

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [UpdateDefaultBranch](#) na Referência de AWS CLI Comandos.

update-pull-request-approval-rule-content

O código de exemplo a seguir mostra como usar `update-pull-request-approval-rule-content`.

AWS CLI

Para editar uma regra de aprovação para um pull request

O `update-pull-request-approval-rule-content` exemplo a seguir atualiza a regra de aprovação especificada por ela para exigir a aprovação de um usuário de um grupo de aprovação que inclui qualquer IAM usuário na 123456789012 AWS conta.

```
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/*\"]}]}"
```

Saída:

```
{  
  "approvalRule": {  
    "approvalRuleContent": "{Version: 2018-11-08, Statements: \  
  [{Type: \"Approvers\", NumberOfApprovalsNeeded: 1, ApprovalPoolMembers: \  
  [\"CodeCommitApprovers:123456789012:user/*\"]}]}",  
    "approvalRuleId": "aac33506-EXAMPLE",  
    "originApprovalRuleTemplate": {},  
    "creationDate": 1570752871.932,  
  },  
}
```

```
"lastModifiedDate": 1570754058.333,  
"approvalRuleName": "Require two approved approvers",  
"lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
"ruleContentSha256": "cd93921cEXAMPLE",  
  }  
}
```

Para obter mais informações, consulte [Editar ou excluir uma regra de aprovação](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [UpdatePullRequestApprovalRuleContent](#) na Referência de AWS CLI Comandos.

update-pull-request-approval-state

O código de exemplo a seguir mostra como usar `update-pull-request-approval-state`.

AWS CLI

Para aprovar ou revogar a aprovação de um pull request

O `update-pull-request-approval-state` exemplo a seguir aprova uma pull request com o ID de 27 e um ID de revisão de `9f29d167EXAMPLE`. Se você quiser revogar a aprovação em vez disso, defina o valor do `--approval-state` parâmetro como `REVOKE`.

```
aws codecommit update-pull-request-approval-state \  
  --pull-request-id 27 \  
  --revision-id 9f29d167EXAMPLE \  
  --approval-state "APPROVE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Revisar uma pull request](#) no Guia AWS CodeCommit do usuário.

- Para API obter detalhes, consulte [UpdatePullRequestApprovalState](#) na Referência de AWS CLI Comandos.

update-pull-request-description

O código de exemplo a seguir mostra como usar `update-pull-request-description`.

AWS CLI

Para alterar a descrição de uma pull request

Este exemplo demonstra como alterar a descrição de uma pull request com o ID de 47.

```
aws codecommit update-pull-request-description \  
  --pull-request-id 47 \  
  --description "Updated the pull request to remove unused global variable."
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [UpdatePullRequestDescription](#) na Referência de AWS CLI Comandos.

update-pull-request-status

O código de exemplo a seguir mostra como usar `update-pull-request-status`.

AWS CLI

Para alterar o status de uma pull request

Este exemplo demonstra como alterar o status de uma pull request com o ID de 42 para um status de `CLOSED` em um AWS CodeCommit repositório chamado. `MyDemoRepo`

```
aws codecommit update-pull-request-status \  
  --pull-request-id 42 \  
  --pull-request-status CLOSED
```

Saída:

```
{  
  "pullRequest": {  
    "approvalRules": [  
      {  
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":  
[{\n\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers\":  
[\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}\"",  
        "approvalRuleId": "dd8b17fe-EXAMPLE",  
        "approvalRuleName": "2-approvers-needed-for-this-change",  
        "creationDate": 1571356106.936,  
        "lastModifiedDate": 571356106.936,  
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",  
        "ruleContentSha256": "4711b576EXAMPLE"  
      }  
    ],  
    "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",  
    "clientRequestToken": "",  
    "creationDate": 1508530823.165,  
    "description": "Updated the pull request to remove unused global variable.",  
    "lastActivityDate": 1508372423.12,  
    "pullRequestId": "47",  
    "pullRequestStatus": "CLOSED",  
    "pullRequestTargets": [  
      {  
        "destinationCommit": "9f31c968EXAMPLE",  
        "destinationReference": "refs/heads/main",
```

```

        "mergeMetadata": {
            "isMerged": false,
        },
        "repositoryName": "MyDemoRepo",
        "sourceCommit": "99132ab0EXAMPLE",
        "sourceReference": "refs/heads/variables-branch"
    }
],
"title": "Consolidation of global variables"
}
}

```

- Para API obter detalhes, consulte [UpdatePullRequestStatus](#) na Referência de AWS CLI Comandos.

update-pull-request-title

O código de exemplo a seguir mostra como usar `update-pull-request-title`.

AWS CLI

Para alterar o título de uma pull request

Este exemplo demonstra como alterar o título de uma pull request com o ID de 47.

```

aws codecommit update-pull-request-title \
  --pull-request-id 47 \
  --title "Consolidation of global variables - updated review"

```

Saída:

```

{
  "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"
}
}

```

- Para API obter detalhes, consulte [UpdatePullRequestTitle](#) na Referência de AWS CLI Comandos.

update-repository-description

O código de exemplo a seguir mostra como usar `update-repository-description`.

AWS CLI

Para alterar a descrição de um repositório

Este exemplo altera a descrição de um AWS CodeCommit repositório. Esse comando gerará saídas somente se houver erros.

Comando:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --  
repository-description "This description was changed"
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [UpdateRepositoryDescription](#) na Referência de AWS CLI Comandos.

update-repository-name

O código de exemplo a seguir mostra como usar `update-repository-name`.

AWS CLI

Para alterar o nome de um repositório

Este exemplo altera o nome de um AWS CodeCommit repositório. Esse comando gerará saídas somente se houver erros. A alteração do nome do AWS CodeCommit repositório mudará o SSH e HTTPS URLs que os usuários precisam para se conectar ao repositório. Os usuários não poderão se conectar ao repositório até que atualizem suas configurações de conexão. Além disso, como o repositório ARN mudará, alterar o nome do repositório invalidará qualquer política de IAM usuário que dependa desse repositório. ARN

Comando:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-  
name MyRenamedDemoRepo
```

Saída:

None .

- Para API obter detalhes, consulte [UpdateRepositoryName](#) na Referência de AWS CLI Comandos.

CodeDeploy exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodeDeploy.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-on-premises-instances

O código de exemplo a seguir mostra como usar `add-tags-to-on-premises-instances`.

AWS CLI

Para adicionar tags às instâncias locais

O `add-tags-to-on-premises-instances` exemplo a seguir associa AWS CodeDeploy a mesma tag de instância local a duas instâncias locais. Ele não registra as instâncias locais com AWS CodeDeploy.

```
aws deploy add-tags-to-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AddTagsToOnPremisesInstances](#) na Referência de AWS CLI Comandos.

batch-get-application-revisions

O código de exemplo a seguir mostra como usar `batch-get-application-revisions`.

AWS CLI

Para recuperar informações sobre revisões de aplicativos

O `batch-get-application-revisions` exemplo a seguir recupera informações sobre a revisão especificada armazenada em um GitHub repositório.

```
aws deploy batch-get-application-revisions \
  --application-name my-codedeploy-application \
  --revisions "[{\\"githubLocation\\": {\\"commitId\\":
  \\"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\\",\\"repository\\": \\"my-github-token/my-
  repository\\"},\\"revisionType\\": \\"GitHub\\"}]"
```

Saída:

```
{
  "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": ""
}
```

Para obter mais informações, consulte [BatchGetApplicationRevisions](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [BatchGetApplicationRevisions](#) na Referência de AWS CLI Comandos.

batch-get-applications

O código de exemplo a seguir mostra como usar `batch-get-applications`.

AWS CLI

Para obter informações sobre vários aplicativos

O `batch-get-applications` exemplo a seguir exibe informações sobre vários aplicativos associados à AWS conta do usuário.

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [BatchGetApplications](#) na Referência de AWS CLI Comandos.

batch-get-deployment-groups

O código de exemplo a seguir mostra como usar batch-get-deployment-groups.

AWS CLI

Para recuperar informações sobre um ou mais grupos de implantação

O batch-get-deployment-groups exemplo a seguir recupera informações sobre dois dos grupos de implantação associados ao CodeDeploy aplicativo especificado.

```
aws deploy batch-get-deployment-groups \  
  --application-name my-codedeploy-application \  
  --deployment-group-names ["my-deployment-group-1","my-deployment-group-2"]
```

Saída:

```
{  
  "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": ""
}

```

Para obter mais informações, consulte [BatchGetDeploymentGroups](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [BatchGetDeploymentGroups](#) na Referência de AWS CLI Comandos.

batch-get-deployment-targets

O código de exemplo a seguir mostra como usar `batch-get-deployment-targets`.

AWS CLI

Para recuperar os alvos associados a uma implantação

O `batch-get-deployment-targets` exemplo a seguir retorna informações sobre um dos destinos associados à implantação especificada.

```

aws deploy batch-get-deployment-targets \
  --deployment-id "d-1A2B3C4D5" \
  --target-ids "i-01a2b3c4d5e6f1111"

```

Saída:

```

{
  "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"
}
]
}
```

Para obter mais informações, consulte [BatchGetDeploymentTargets](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [BatchGetDeploymentTargets](#) na Referência de AWS CLI Comandos.

batch-get-deployments

O código de exemplo a seguir mostra como usar batch-get-deployments.

AWS CLI

Para obter informações sobre várias implantações

O batch-get-deployments exemplo a seguir exibe informações sobre várias implantações associadas à AWS conta do usuário.

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

Saída:

```
{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [BatchGetDeployments](#) na Referência de AWS CLI Comandos.

batch-get-on-premises-instances

O código de exemplo a seguir mostra como usar `batch-get-on-premises-instances`.

AWS CLI

Para obter informações sobre uma ou mais instâncias locais

O `batch-get-on-premises-instances` exemplo a seguir obtém informações sobre duas instâncias locais.

```
aws deploy batch-get-on-premises-instances --instance-  
names AssetTag12010298EX AssetTag23121309EX
```

Saída:

```
{  
  "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"
    }
]
}
```

- Para API obter detalhes, consulte [BatchGetOnPremisesInstances](#) na Referência de AWS CLI Comandos.

continue-deployment

O código de exemplo a seguir mostra como usar `continue-deployment`.

AWS CLI

Para começar a redirecionar o tráfego sem esperar que um tempo de espera especificado termine.

O `continue-deployment` exemplo a seguir começa a redirecionar o tráfego de instâncias no ambiente original que estão prontas para começar a transferir o tráfego para instâncias no ambiente substituto.

```
aws deploy continue-deployment \
  --deployment-id "d-A1B2C3111" \
  --deployment-wait-type "READY_WAIT"
```

Este comando não produz saída.

Para obter mais informações, consulte [ContinueDeployment](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [ContinueDeployment](#) na Referência de AWS CLI Comandos.

create-application

O código de exemplo a seguir mostra como usar `create-application`.

AWS CLI

Para criar um aplicativo

O `create-application` exemplo a seguir cria um aplicativo e o associa à AWS conta do usuário.

```
aws deploy create-application --application-name MyOther_App
```

Saída:

```
{
  "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- Para API obter detalhes, consulte [CreateApplication](#) na Referência de AWS CLI Comandos.

create-deployment-config

O código de exemplo a seguir mostra como usar `create-deployment-config`.

AWS CLI

Para criar uma configuração de implantação personalizada

O `create-deployment-config` exemplo a seguir cria uma configuração de implantação personalizada e a associa à AWS conta do usuário.

```
aws deploy create-deployment-config \
  --deployment-config-name ThreeQuartersHealthy \
  --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

Saída:

```
{
  "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- Para API obter detalhes, consulte [CreateDeploymentConfig](#) na Referência de AWS CLI Comandos.

create-deployment-group

O código de exemplo a seguir mostra como usar `create-deployment-group`.

AWS CLI

Para criar um grupo de implantação

O `create-deployment-group` exemplo a seguir cria um grupo de implantação e o associa ao aplicativo especificado e à AWS conta do usuário.

```
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
```

Saída:

```
{  
  "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- Para API obter detalhes, consulte [CreateDeploymentGroup](#) na Referência de AWS CLI Comandos.

create-deployment

O código de exemplo a seguir mostra como usar `create-deployment`.

AWS CLI

Exemplo 1: Para criar uma CodeDeploy implantação usando a plataforma de computação EC2 / On-premises

O `create-deployment` exemplo a seguir cria uma implantação e a associa à AWS conta do usuário.

```
aws deploy create-deployment \  
  --application-name WordPress_App \  
  --deployment-config-name CodeDeployDefault.OneAtATime \  
  --deployment-group-name WordPress_DG \  
  --description "My demo deployment" \  
  --service-role-arn arn:aws:iam::123456789012:role/CodeDeployDemoRole
```

--s3-**location** *bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke*

Saída:

```
{
  "deploymentId": "d-A1B2C3111"
}
```

Exemplo 2: Para criar uma CodeDeploy implantação usando a plataforma de ECS computação da Amazon

O `create-deployment` exemplo a seguir usa os dois arquivos a seguir para implantar um ECS serviço da Amazon.

Conteúdo do arquivo `create-deployment.json`:

```
{
  "applicationName": "ecs-deployment",
  "deploymentGroupName": "ecs-deployment-dg",
  "revision": {
    "revisionType": "S3",
    "s3Location": {
      "bucket": "ecs-deployment-bucket",
      "key": "appspec.yaml",
      "bundleType": "YAML"
    }
  }
}
```

Esse arquivo, por sua vez, recupera o seguinte arquivo `appspec.yaml` de um bucket do S3 chamado `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
```

Saída:

```
{
  "deploymentId": "d-1234ABCDE"
}
```

Para obter mais informações, consulte [CreateDeployment](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [CreateDeployment](#) na Referência de AWS CLI Comandos.

delete-application

O código de exemplo a seguir mostra como usar delete-application.

AWS CLI

Como excluir uma aplicação

O delete-application exemplo a seguir exclui o aplicativo especificado que está associado à AWS conta do usuário.

```
aws deploy delete-application --application-name WordPress_App
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteApplication](#) na Referência de AWS CLI Comandos.

delete-deployment-config

O código de exemplo a seguir mostra como usar delete-deployment-config.

AWS CLI

Para excluir uma configuração de implantação

O `delete-deployment-config` exemplo a seguir exclui uma configuração de implantação personalizada associada à AWS conta do usuário.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteDeploymentConfig](#) na Referência de AWS CLI Comandos.

`delete-deployment-group`

O código de exemplo a seguir mostra como usar `delete-deployment-group`.

AWS CLI

Para excluir um grupo de implantação

O `delete-deployment-group` exemplo a seguir exclui um grupo de implantação associado ao aplicativo especificado.

```
aws deploy delete-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DG
```

Saída:

```
{  
  "hooksNotCleanedUp": []  
}
```

- Para API obter detalhes, consulte [DeleteDeploymentGroup](#) na Referência de AWS CLI Comandos.

`delete-git-hub-account-token`

O código de exemplo a seguir mostra como usar `delete-git-hub-account-token`.

AWS CLI

Para excluir uma conexão de GitHub conta

O `delete-git-hub-account-token` exemplo a seguir exclui a conexão da GitHub conta especificada.

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

Saída:

```
{
  "tokenName": "my-github-account"
}
```

Para obter mais informações, consulte [DeleteGitHubAccountToken](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [DeleteGitHubAccountToken](#) na Referência de AWS CLI Comandos.

deregister-on-premises-instance

O código de exemplo a seguir mostra como usar `deregister-on-premises-instance`.

AWS CLI

Para cancelar o registro de uma instância local

O `deregister-on-premises-instance` exemplo a seguir cancela o registro de uma instância local com AWS CodeDeploy, mas não exclui o IAM usuário associado à instância nem se desassocia da instância nas tags AWS CodeDeploy da instância local. Ele também não desinstala o AWS CodeDeploy Agente da instância nem remove o arquivo de configuração local da instância.

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeregisterOnPremisesInstance](#) na Referência de AWS CLI Comandos.

deregister

O código de exemplo a seguir mostra como usar `deregister`.

AWS CLI

Para cancelar o registro de uma instância local

O `deregister` exemplo a seguir cancela o registro de uma instância local com. AWS CodeDeploy Ela não exclui o IAM usuário associado à instância. Ele se dissocia AWS CodeDeploy das tags locais da instância. Ele não desinstala o AWS CodeDeploy Agente da instância nem remove o arquivo de configuração local da instância.

```
aws deploy deregister \  
  --instance-name AssetTag12010298EX \  
  --no-delete-iam-user \  
  --region us-west-2
```

Saída:

```
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
```

- Para API obter detalhes, consulte [Cancelar o registro na Referência](#) de AWS CLI Comandos.

get-application-revision

O código de exemplo a seguir mostra como usar `get-application-revision`.

AWS CLI

Para obter informações sobre uma revisão do aplicativo

O `get-application-revision` exemplo a seguir exibe informações sobre uma revisão do aplicativo associada ao aplicativo especificado.

```
aws deploy get-application-revision \  
  --application-name WordPress_App \  
  --s3-  
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetApplicationRevision](#) na Referência de AWS CLI Comandos.

get-application

O código de exemplo a seguir mostra como usar `get-application`.

AWS CLI

Para obter informações sobre um aplicativo

O `get-application` exemplo a seguir exibe informações sobre um aplicativo associado à AWS conta do usuário.

```
aws deploy get-application --application-name WordPress_App
```

Saída:

```
{
  "application": {
    "applicationName": "WordPress_App",
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "createTime": 1407878168.078,
    "linkedToGitHub": false
  }
}
```

- Para API obter detalhes, consulte [GetApplication](#) na Referência de AWS CLI Comandos.

get-deployment-config

O código de exemplo a seguir mostra como usar `get-deployment-config`.

AWS CLI

Para obter informações sobre uma configuração de implantação

O `get-deployment-config` exemplo a seguir exibe informações sobre uma configuração de implantação associada à AWS conta do usuário.

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Saída:

```
{
  "deploymentConfigInfo": {
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "minimumHealthyHosts": {
      "type": "FLEET_PERCENT",
      "value": 75
    },
    "createTime": 1411081164.379,
    "deploymentConfigName": "ThreeQuartersHealthy"
  }
}
```

```
}
```

- Para API obter detalhes, consulte [GetDeploymentConfig](#) na Referência de AWS CLI Comandos.

get-deployment-group

O código de exemplo a seguir mostra como usar `get-deployment-group`.

AWS CLI

Para ver informações sobre um grupo de implantação

O `get-deployment-group` exemplo a seguir exibe informações sobre um grupo de implantação associado ao aplicativo especificado.

```
aws deploy get-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DG
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [GetDeploymentGroup](#) na Referência de AWS CLI Comandos.

get-deployment-instance

O código de exemplo a seguir mostra como usar `get-deployment-instance`.

AWS CLI

Para obter informações sobre uma instância de implantação

O `get-deployment-instance` exemplo a seguir exibe informações sobre uma instância de implantação associada à implantação especificada.

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-id i-902e9fEX
```

Saída:

```
{
  "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"
}
}

```

- Para API obter detalhes, consulte [GetDeploymentInstance](#) na Referência de AWS CLI Comandos.

get-deployment-target

O código de exemplo a seguir mostra como usar `get-deployment-target`.

AWS CLI

Para retornar informações sobre um alvo de implantação

O `get-deployment-target` exemplo a seguir retorna informações sobre um destino de implantação associado à implantação especificada.

```

aws deploy get-deployment-target \
  --deployment-id "d-A1B2C3111" \
  --target-id "i-a1b2c3d4e5f61111"

```

Saída:

```
{
  "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"
}
}
```

Para obter mais informações, consulte [GetDeploymentTarget](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [GetDeploymentTarget](#) na Referência de AWS CLI Comandos.

get-deployment

O código de exemplo a seguir mostra como usar `get-deployment`.

AWS CLI

Para obter informações sobre uma implantação

O `get-deployment` exemplo a seguir exibe informações sobre uma implantação associada à AWS conta do usuário.

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

Saída:

```
{
  "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
}
}
```

- Para API obter detalhes, consulte [GetDeployment](#) na Referência de AWS CLI Comandos.

get-on-premises-instance

O código de exemplo a seguir mostra como usar `get-on-premises-instance`.

AWS CLI

Para obter informações sobre uma instância local

O `get-on-premises-instance` exemplo a seguir recupera informações sobre a instância local especificada.

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Saída:

```
{
  "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"
  }
}

```

- Para API obter detalhes, consulte [GetOnPremisesInstance](#) na Referência de AWS CLI Comandos.

install

O código de exemplo a seguir mostra como usar `install`.

AWS CLI

Para instalar uma instância local

O `install` exemplo a seguir copia o arquivo de configuração local do local especificado na instância para o local na instância em que o AWS CodeDeploy agente espera encontrá-lo. Ele também instala o AWS CodeDeploy Agente na instância. Ele não cria nenhum IAM usuário, nem registra a instância local nem associa nenhuma tag de instância local AWS CodeDeploy à instância. 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

```

Saída:

```

Creating the on-premises instance configuration file... DONE

```

```
Installing the AWS CodeDeploy Agent... DONE
```

- Para API obter detalhes, consulte [Instalar](#) na Referência de AWS CLI Comandos.

list-application-revisions

O código de exemplo a seguir mostra como usar `list-application-revisions`.

AWS CLI

Para obter informações sobre revisões de aplicativos

O `list-application-revisions` exemplo a seguir exibe informações sobre todas as revisões do aplicativo associadas ao aplicativo especificado.

```
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
```

Saída:

```
{
  "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"
      }
    ]
  }
```

- Para API obter detalhes, consulte [ListApplicationRevisions](#) na Referência de AWS CLI Comandos.

list-applications

O código de exemplo a seguir mostra como usar `list-applications`.

AWS CLI

Para obter informações sobre aplicativos

O `list-applications` exemplo a seguir exibe informações sobre todos os aplicativos associados à AWS conta do usuário.

```
aws deploy list-applications
```

Saída:

```
{
  "applications": [
    "WordPress_App",
    "MyOther_App"
  ]
}
```

- Para API obter detalhes, consulte [ListApplications](#) na Referência de AWS CLI Comandos.

list-deployment-configs

O código de exemplo a seguir mostra como usar `list-deployment-configs`.

AWS CLI

Para obter informações sobre as configurações de implantação

O `list-deployment-configs` exemplo a seguir exibe informações sobre todas as configurações de implantação associadas à AWS conta do usuário.

```
aws deploy list-deployment-configs
```

Saída:

```
{
  "deploymentConfigsList": [
    "ThreeQuartersHealthy",
    "CodeDeployDefault.AllAtOnce",
    "CodeDeployDefault.HalfAtATime",
    "CodeDeployDefault.OneAtATime"
  ]
}
```

- Para API obter detalhes, consulte [ListDeploymentConfigs](#) na Referência de AWS CLI Comandos.

list-deployment-groups

O código de exemplo a seguir mostra como usar `list-deployment-groups`.

AWS CLI

Para obter informações sobre grupos de implantação

O `list-deployment-groups` exemplo a seguir exibe informações sobre todos os grupos de implantação associados ao aplicativo especificado.

```
aws deploy list-deployment-groups --application-name WordPress_App
```

Saída:

```
{
  "applicationName": "WordPress_App",
  "deploymentGroups": [
    "WordPress_DG",
    "WordPress_Beta_DG"
  ]
}
```

- Para API obter detalhes, consulte [ListDeploymentGroups](#) na Referência de AWS CLI Comandos.

list-deployment-instances

O código de exemplo a seguir mostra como usar `list-deployment-instances`.

AWS CLI

Para obter informações sobre instâncias de implantação

O `list-deployment-instances` exemplo a seguir exibe informações sobre todas as instâncias de implantação associadas à implantação especificada.

```
aws deploy list-deployment-instances \  
  --deployment-id d-A1B2C3111 \  
  --instance-status-filter Succeeded
```

Saída:

```
{  
  "instancesList": [  
    "i-EXAMPLE11",  
    "i-EXAMPLE22"  
  ]  
}
```

- Para API obter detalhes, consulte [ListDeploymentInstances](#) na Referência de AWS CLI Comandos.

list-deployment-targets

O código de exemplo a seguir mostra como usar `list-deployment-targets`.

AWS CLI

Para recuperar uma lista de destinos associados IDs a uma implantação

O `list-deployment-targets` exemplo a seguir recupera uma lista de destinos IDs associados às implantações que têm o status “Falha” ou “InProgress”.

```
aws deploy list-deployment-targets \  
  --deployment-id "d-A1B2C3111" \  
  --target-filters "{\"TargetStatus\": [\"Failed\", \"InProgress\"]}"
```

Saída:

```
{  
  "targetIds": [  
    "i-0f1558aaf90e5f1f9"  
  ]  
}
```

Para obter mais informações, consulte [ListDeploymentTargets](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [ListDeploymentTargets](#) na Referência de AWS CLI Comandos.

list-deployments

O código de exemplo a seguir mostra como usar `list-deployments`.

AWS CLI

Para obter informações sobre implantações

O `list-deployments` exemplo a seguir exibe informações sobre todas as implantações associadas ao aplicativo e ao grupo de implantação especificados.

```
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
```

Saída:

```
{  
  "deployments": [  
    "d-EXAMPLE11",  
    "d-EXAMPLE22",
```

```
    "d-EXAMPLE33"  
  ]  
}
```

- Para API obter detalhes, consulte [ListDeployments](#) na Referência de AWS CLI Comandos.

list-git-hub-account-token-names

O código de exemplo a seguir mostra como usar `list-git-hub-account-token-names`.

AWS CLI

Para listar os nomes das conexões armazenadas com as GitHub contas

O `list-git-hub-account-token-names` exemplo a seguir lista os nomes das conexões armazenadas com as GitHub contas do AWS usuário atual.

```
aws deploy list-git-hub-account-token-names
```

Saída:

```
{  
  "tokenNameList": [  
    "my-first-token",  
    "my-second-token",  
    "my-third-token"  
  ]  
}
```

Para obter mais informações, consulte [ListGitHubAccountTokenNames](#) na AWS CodeDeploy API Referência.

- Para API obter detalhes, consulte [ListGitHubAccountTokenNames](#) na Referência de AWS CLI Comandos.

list-on-premises-instances

O código de exemplo a seguir mostra como usar `list-on-premises-instances`.

AWS CLI

Para obter informações sobre uma ou mais instâncias locais

O `list-on-premises-instances` exemplo a seguir recupera uma lista de nomes de instâncias locais disponíveis para instâncias registradas AWS CodeDeploy e que também têm a tag de instância local especificada associada à instância. AWS CodeDeploy

```
aws deploy list-on-premises-instances \  
  --registration-status Registered \  
  --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

Saída:

```
{  
  "instanceNames": [  
    "AssetTag12010298EX"  
  ]  
}
```

- Para API obter detalhes, consulte [ListOnPremisesInstances](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um recurso (aplicativo)

O `list-tags-for-resource` exemplo a seguir lista as tags aplicadas a um aplicativo chamado `testApp` em CodeDeploy.

```
aws deploy list-tags-for-resource \  
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Type",  
      "Value": "testType"    }  
  ]  
}
```

```

    },
    {
      "Key": "Name",
      "Value": "testName"
    }
  ]
}

```

Para obter mais informações, consulte Como [marcar instâncias para grupos de implantação CodeDeploy no](#) Guia do AWS CodeDeploy usuário.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

push

O código de exemplo a seguir mostra como usar push.

AWS CLI

Para agrupar e implantar uma revisão de aplicativo AWS CodeDeploy compatível no Amazon S3

O push exemplo a seguir agrupa e implanta uma revisão do aplicativo no Amazon S3 e, em seguida, associa a revisão do aplicativo ao aplicativo especificado.

```

aws deploy push \
  --application-name WordPress_App \
  --description "This is my deployment" \
  --ignore-hidden-files \
  --s3-location s3://CodeDeployDemoBucket/WordPressApp.zip \
  --source /tmp/MyLocalDeploymentFolder/

```

A saída descreve como usar o create-deployment comando para criar uma implantação que usa a revisão do aplicativo carregada.

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

```

- Para API obter detalhes, consulte [Push](#) in AWS CLI Command Reference.

register-application-revision

O código de exemplo a seguir mostra como usar `register-application-revision`.

AWS CLI

Para registrar informações sobre uma revisão de aplicativo já carregada

O `register-application-revision` exemplo a seguir registra informações sobre uma revisão de aplicativo já carregada armazenada no Amazon S3 com. 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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RegisterApplicationRevision](#) na Referência de AWS CLI Comandos.

register-on-premises-instance

O código de exemplo a seguir mostra como usar `register-on-premises-instance`.

AWS CLI

Para registrar uma instância local

O `register-on-premises-instance` exemplo a seguir registra uma instância local com. AWS CodeDeploy Ele não cria o IAM usuário especificado nem associa tags à instância registrada em AWS CodeDeploy nenhuma instância local.

```
aws deploy register-on-premises-instance \  
  --instance-name AssetTag12010298EX \  
  --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RegisterOnPremisesInstance](#) na Referência de AWS CLI Comandos.

register

O código de exemplo a seguir mostra como usar `register`.

AWS CLI

Para registrar uma instância local

O `register` exemplo a seguir registra uma instância local com AWS CodeDeploy, AWS CodeDeploy associa na tag de instância local especificada à instância registrada e cria um arquivo de configuração local que pode ser copiado para a instância. Ele não cria o IAM usuário nem instala o AWS CodeDeploy Agente na instância.

```
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
```

Saída:

```
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
```

- Para API obter detalhes, consulte [Registrar](#) na Referência de AWS CLI Comandos.

remove-tags-from-on-premises-instances

O código de exemplo a seguir mostra como usar `remove-tags-from-on-premises-instances`.

AWS CLI

Para remover tags de uma ou mais instâncias locais

O `remove-tags-from-on-premises-instances` exemplo a seguir dissocia as tags locais especificadas das instâncias locais. AWS CodeDeploy Ele não cancela o registro das instâncias locais AWS CodeDeploy, nem desinstala o AWS CodeDeploy Agente da instância, nem remove

o arquivo de configuração local das instâncias, nem exclui os IAM usuários associados às instâncias.

```
aws deploy remove-tags-from-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RemoveTagsFromOnPremisesInstances](#) na Referência de AWS CLI Comandos.

stop-deployment

O código de exemplo a seguir mostra como usar stop-deployment.

AWS CLI

Para tentar interromper uma implantação

O stop-deployment exemplo a seguir tenta interromper uma implantação em andamento associada à AWS conta do usuário.

```
aws deploy stop-deployment --deployment-id d-a1b2c3111
```

Saída:

```
{  
  "status": "Succeeded",  
  "statusMessage": "No more commands will be scheduled for execution in the  
  deployment instances"  
}
```

- Para API obter detalhes, consulte [StopDeployment](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar um recurso (aplicativo)

O `tag-resource` exemplo a seguir adiciona duas tags com as chaves `Name` e `Type` e values `testName` e `testType` a um aplicativo chamado `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
```

Se for bem-sucedido, esse comando não gerará nenhum resultado.

Para obter mais informações, consulte Como [marcar instâncias para grupos de implantação CodeDeploy no](#) Guia do AWS CodeDeploy usuário.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

uninstall

O código de exemplo a seguir mostra como usar `uninstall`.

AWS CLI

Para desinstalar uma instância local

O `uninstall` exemplo a seguir desinstala o AWS CodeDeploy Agente da instância local e remove o arquivo de configuração local da instância. Ele não cancela o registro da instância AWS CodeDeploy, nem desassocia nenhuma tag de instância local AWS CodeDeploy da instância, nem exclui o IAM usuário associado à instância.

```
aws deploy uninstall
```

Este comando não produz saída.

- Para API obter detalhes, consulte [Desinstalar](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um recurso (aplicativo)

O `untag-resource` exemplo a seguir remove duas tags com as chaves `Name` e `Type` de um aplicativo chamado `testApp` in CodeDeploy.

```
aws deploy untag-resource \  
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp \  
  --tag-keys Name Type
```

Se for bem-sucedido, esse comando não gerará nenhum resultado.

Para obter mais informações, consulte Como [marcar instâncias para grupos de implantação CodeDeploy no](#) Guia do AWS CodeDeploy usuário.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-application

O código de exemplo a seguir mostra como usar `update-application`.

AWS CLI

Para alterar os detalhes de um aplicativo

O `update-application` exemplo a seguir altera o nome de um aplicativo associado à AWS conta do usuário.

```
aws deploy update-application \  
  --application-name WordPress_App \  
  --new-application-name My_WordPress_App
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateApplication](#) na Referência de AWS CLI Comandos.

update-deployment-group

O código de exemplo a seguir mostra como usar `update-deployment-group`.

AWS CLI

Para alterar as informações sobre um grupo de implantação

O `update-deployment-group` exemplo a seguir altera as configurações de um grupo de implantação associado ao aplicativo especificado.

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateDeploymentGroup](#) na Referência de AWS CLI Comandos.

CodeGuru Exemplos de revisores usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodeGuru Reviewer.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-repository

O código de exemplo a seguir mostra como usar `associate-repository`.

AWS CLI

Exemplo 1: Para criar uma associação de repositório do Bitbucket

O `associate-repository` exemplo a seguir cria uma associação de repositório usando um repositório Bitbucket existente.

```
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 }'
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criar uma associação de repositório Bitbucket no Amazon CodeGuru Reviewer no Guia do usuário](#) do Amazon CodeGuru Reviewer.

Exemplo 2: Para criar uma associação de repositório GitHub corporativo

O `associate-repository` exemplo a seguir cria uma associação de repositório usando um repositório GitHub corporativo existente.

```
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 }'
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Criar uma associação de repositório do GitHub Enterprise Server no Amazon CodeGuru Reviewer no Guia](#) do usuário do Amazon Codeguru Reviewer.

Exemplo 3: Para criar uma associação de AWS CodeCommit repositório

O `associate-repository` exemplo a seguir cria uma associação de repositório usando um AWS CodeCommit repositório existente.

```
aws codeguru-reviewer associate-repository \
  --repository CodeCommit={Name=mySampleRepo}
```

Saída:

```
{
  "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"
```

```
}
}
```

Para obter mais informações, consulte [Criar uma associação de AWS CodeCommit repositório no Amazon CodeGuru Reviewer no Guia](#) do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [AssociateRepository](#) na Referência de AWS CLI Comandos.

create-code-review

O código de exemplo a seguir mostra como usar `create-code-review`.

AWS CLI

Para criar uma revisão de código.

O seguinte `create-code-review` cria uma revisão do código na `mainline` ramificação de um AWS CodeCommit repositório chamado `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"}}}'
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Criar análises de código no Amazon CodeGuru Reviewer](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [CreateCodeReview](#) na Referência de AWS CLI Comandos.

describe-code-review

O código de exemplo a seguir mostra como usar `describe-code-review`.

AWS CLI

Listar detalhes sobre uma revisão de código.

A seguir `describe-code-review` estão listadas informações sobre uma revisão do código na ramificação “principal” de um AWS CodeCommit repositório chamado “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-abcdefgh12345678 \
  --recommendation-
id 3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb \
  --reactions ThumbsUp

```

Saída

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Exibir detalhes da revisão de código](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [DescribeCodeReview](#) na Referência de AWS CLI Comandos.

describe-recommendation-feedback

O código de exemplo a seguir mostra como usar `describe-recommendation-feedback`.

AWS CLI

Para ver informações sobre feedback sobre uma recomendação

A seguir, são `describe-recommendation-feedback` exibidas informações sobre comentários sobre uma recomendação. Essa recomendação tem uma `ThumbsUp` reação.

```

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

```

Saída:

```
{
```

```

    "RecommendationFeedback": {
      "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefg12345678",
      "RecommendationId":
"3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
      "Reactions": [
        "ThumbsUp"
      ],
      "UserId": "aws-user-id",
      "CreatedTimeStamp": 1618877070.313,
      "LastUpdatedTimeStamp": 1618877948.881
    }
  }
}

```

Para obter mais informações, consulte [Visualizar recomendações e fornecer feedback](#) e [Etapa 4: Fornecer feedback](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [DescribeRecommendationFeedback](#) na Referência de AWS CLI Comandos.

describe-repository-association

O código de exemplo a seguir mostra como usar `describe-repository-association`.

AWS CLI

Exemplo 1: Para retornar informações sobre uma associação de GitHub repositório

O `describe-repository-association` exemplo a seguir retorna informações sobre uma associação de repositório que usa um repositório GitHub corporativo e está no Associated estado.

```

aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Criar uma associação de repositório do GitHub Enterprise Server no Amazon CodeGuru Reviewer no Guia](#) do usuário do Amazon CodeGuru Reviewer.

Exemplo 2: Para retornar informações sobre uma associação de repositório com falha

O `describe-repository-association` exemplo a seguir retorna informações sobre uma associação de repositório que usa um repositório GitHub corporativo e está no `Failed` estado.

```

aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Criar uma associação de repositório do GitHub Enterprise Server no Amazon CodeGuru Reviewer no Guia](#) do usuário do Amazon CodeGuru Reviewer.

Exemplo 3: Para retornar informações sobre uma associação de repositório desassociada

O `describe-repository-association` exemplo a seguir retorna informações sobre uma associação de repositório que usa um repositório GitHub corporativo e está no `Disassociating` estado.

```
aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Criar uma associação de repositório do GitHub Enterprise Server no Amazon CodeGuru Reviewer no Guia](#) do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [DescribeRepositoryAssociation](#) na Referência de AWS CLI Comandos.

disassociate-repository

O código de exemplo a seguir mostra como usar `disassociate-repository`.

AWS CLI

Para desassociar uma associação de repositório

O seguinte `disassociate-repository` dissocia uma associação de repositório que está usando um repositório. AWS CodeCommit

```
aws codeguru-reviewer disassociate-repository \  
  --association-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Desassociar um repositório no CodeGuru Reviewer no Amazon CodeGuru Reviewer User Guide](#).

- Para API obter detalhes, consulte [DisassociateRepository](#) na Referência de AWS CLI Comandos.

list-code-reviews

O código de exemplo a seguir mostra como usar `list-code-reviews`.

AWS CLI

Para listar as avaliações de código criadas em sua AWS conta nos últimos 90 dias.

O `list-code-reviews` exemplo a seguir lista as análises de código criadas nos últimos 90 dias usando pull requests.

```
aws codeguru-reviewer list-code-reviews \  
  --type PullRequest
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [Exibir todas as análises de código](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [ListCodeReviews](#) na Referência de AWS CLI Comandos.

list-recommendation-feedback

O código de exemplo a seguir mostra como usar `list-recommendation-feedback`.

AWS CLI

Para listar o feedback de recomendação do cliente para uma recomendação em um repositório associado

A seguir, `list-recommendation-feedback` listamos os comentários dos clientes sobre todas as recomendações sobre uma revisão de código. Essa análise de código tem um feedback, um "ThumbsUp", de um 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
```

Saída:

```
{
  "RecommendationFeedbackSummaries": [
    {
      "RecommendationId":
      "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
      "Reactions": [
        "ThumbsUp"
      ],
      "UserId": "aws-user-id"
    }
  ]
}
```

Para obter mais informações, consulte [Etapa 4: Forneça feedback](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [ListRecommendationFeedback](#) na Referência de AWS CLI Comandos.

list-recommendations

O código de exemplo a seguir mostra como usar `list-recommendations`.

AWS CLI

Para listar as recomendações para uma revisão de código completa

O `list-recommendations` exemplo a seguir lista as recomendações para uma revisão de código completa. Essa análise de código tem uma recomendação.

```
aws codeguru-reviewer list-recommendations \
  --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-
  review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Etapa 4: Forneça feedback](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [ListRecommendations](#) na Referência de AWS CLI Comandos.

list-repository-associations

O código de exemplo a seguir mostra como usar `list-repository-associations`.

AWS CLI

Para listar as associações de repositórios em sua conta AWS

O `list-repository-associations` exemplo a seguir retorna uma lista de objetos de resumo da associação de repositórios em sua conta. Você pode filtrar a lista retornada por `ProviderType` `Name` `State`, `Owner` e.

```
aws codeguru-reviewer list-repository-associations
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Visualizar todas as associações de repositórios no CodeGuru Reviewer no Amazon CodeGuru Reviewer User Guide](#).

- Para API obter detalhes, consulte [ListRepositoryAssociations](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags em um repositório associado

A seguir, `list-tags-for-resource` listamos as tags em um repositório associado. Esse repositório associado tem duas tags.

```
aws codeguru-reviewer list-tags-for-resource \  
  --resource-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "Tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

Para obter mais informações, consulte [Exibir tags para um repositório associado ao CodeGuru Reviewer \(AWS CLI\)](#) no Amazon CodeGuru Reviewer User Guide.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

`put-recommendation-feedback`

O código de exemplo a seguir mostra como usar `put-recommendation-feedback`.

AWS CLI

Para adicionar uma recomendação a uma revisão de código

O texto a seguir `put-recommendation-feedback` apresenta uma ThumbsUp recomendação sobre uma revisão de código.

```
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 \  
  --text Great work!
```

```
--reactions ThumbsUp
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 4: Forneça feedback](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [PutRecommendationFeedback](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a um repositório associado

O seguinte `tag-resource` adiciona duas tags a um repositório associado

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma tag a um repositório associado ao CodeGuru revisor \(AWS CLI\)](#) e [Adicionar ou atualizar tags para um repositório associado ao CodeGuru revisor \(AWS CLI\)](#) no Guia do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para desmarcar um repositório associado

O seguinte `untag-resource` remove duas tags com as chaves “Secret” e “Team” de um repositório associado.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Remover tags de um CodeGuru repositório associado ao revisor \(AWS CLI\) no Guia](#) do usuário do Amazon CodeGuru Reviewer.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

CodePipeline exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodePipeline.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

acknowledge-job

O código de exemplo a seguir mostra como usar `acknowledge-job`.

AWS CLI

Para recuperar informações sobre um trabalho especificado

Este exemplo retorna informações sobre um trabalho especificado, incluindo o status desse trabalho, se ele existir. Isso é usado somente para trabalhadores e ações personalizadas. Para determinar o valor de nonce e o ID do trabalho, use `aws poll-for-jobs codepipeline`.

Comando:

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

Saída:

```
{  
  "status": "InProgress"  
}
```

- Para API obter detalhes, consulte [AcknowledgeJob](#) na Referência de AWS CLI Comandos.

create-custom-action-type

O código de exemplo a seguir mostra como usar `create-custom-action-type`.

AWS CLI

Para criar uma ação personalizada

Este exemplo cria uma ação personalizada para AWS CodePipeline usar um JSON arquivo já criado (aqui denominado `MyCustomAction.json`) que contém a estrutura da ação personalizada. Para obter mais informações sobre os requisitos para criar uma ação personalizada, incluindo a estrutura do arquivo, consulte o Guia AWS CodePipeline do usuário.

```
aws codepipeline create-custom-action-type --cli-input-json file://  
MyCustomAction.json
```

Conteúdo do JSON arquivo `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
}
}
```

Esse comando retorna a estrutura da ação personalizada.

- Para API obter detalhes, consulte [CreateCustomActionType](#) na Referência de AWS CLI Comandos.

create-pipeline

O código de exemplo a seguir mostra como usar create-pipeline.

AWS CLI

Para criar um pipeline

Este exemplo cria um pipeline AWS CodePipeline usando um JSON arquivo já criado (aqui denominado MySecondPipeline .json) que contém a estrutura do pipeline. Para obter mais informações sobre os requisitos para criar um pipeline, incluindo a estrutura do arquivo, consulte o Guia AWS CodePipeline do usuário.

Comando:

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

JSON conteúdo de amostra do arquivo:

```
{
  "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
}
}
```

Saída:

This command returns the structure of the pipeline.

- Para API obter detalhes, consulte [CreatePipeline](#) na Referência de AWS CLI Comandos.

delete-custom-action-type

O código de exemplo a seguir mostra como usar `delete-custom-action-type`.

AWS CLI

Para excluir uma ação personalizada

Este exemplo exclui uma ação personalizada AWS CodePipeline usando um JSON arquivo já criado (aqui chamado `DeleteMyCustomAction.json`) que contém o tipo de ação, o nome do provedor e o número da versão da ação a ser excluída. Use o `list-action-types` comando para visualizar os valores corretos para categoria, versão e provedor.

Comando:

```
aws codepipeline delete-custom-action-type --cli-input-json file://DeleteMyCustomAction.json
```

JSON conteúdo de amostra do arquivo:

```
{
  "category": "Build",
  "version": "1",
  "provider": "MyJenkinsProviderName"
}
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DeleteCustomActionType](#) na Referência de AWS CLI Comandos.

delete-pipeline

O código de exemplo a seguir mostra como usar delete-pipeline.

AWS CLI

Para excluir um pipeline

Este exemplo exclui um pipeline chamado MySecondPipeline de AWS CodePipeline. Use o comando list-pipelines para ver uma lista de pipelines associados à sua conta. AWS

Comando:

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DeletePipeline](#) na Referência de AWS CLI Comandos.

delete-webhook

O código de exemplo a seguir mostra como usar delete-webhook.

AWS CLI

Para excluir um webhook

O delete-webhook exemplo a seguir exclui um webhook para uma ação de origem da GitHub versão 1. Você deve usar o deregister-webhook-with-third-party comando para cancelar o registro do webhook antes de excluí-lo.

```
aws codepipeline delete-webhook \  
  --name my-webhook
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir o webhook da sua GitHub fonte](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [DeleteWebhook](#) na Referência de AWS CLI Comandos.

deregister-webhook-with-third-party

O código de exemplo a seguir mostra como usar deregister-webhook-with-third-party.

AWS CLI

Para cancelar o registro de um webhook

O deregister-webhook-with-third-party exemplo a seguir exclui um webhook para uma ação de origem da GitHub versão 1. É necessário cancelar o registro do webhook antes de excluí-lo.

```
aws codepipeline deregister-webhook-with-third-party \  
  --webhook-name my-webhook
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir o webhook da sua GitHub fonte](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [DeregisterWebhookWithThirdParty](#) na Referência de AWS CLI Comandos.

disable-stage-transition

O código de exemplo a seguir mostra como usar `disable-stage-transition`.

AWS CLI

Para desativar uma transição para um estágio em um pipeline

Este exemplo desativa as transições para o estágio Beta do MyFirstPipeline pipeline em AWS CodePipeline

Comando:

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DisableStageTransition](#) na Referência de AWS CLI Comandos.

enable-stage-transition

O código de exemplo a seguir mostra como usar `enable-stage-transition`.

AWS CLI

Para permitir a transição para um estágio em um pipeline

Este exemplo permite transições para o estágio Beta do MyFirstPipeline pipeline em AWS CodePipeline.

Comando:

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

Saída:

None.

- Para API obter detalhes, consulte [EnableStageTransition](#) na Referência de AWS CLI Comandos.

get-job-details

O código de exemplo a seguir mostra como usar `get-job-details`.

AWS CLI

Para obter detalhes de um trabalho

Este exemplo retorna detalhes sobre um trabalho cujo ID é representado por `f4f4ff82-2d11-EXAMPLE`. Este comando é usado somente para ações personalizadas. Quando esse comando é chamado, AWS CodePipeline retorna credenciais temporárias para o bucket Amazon S3 usado para armazenar artefatos para o pipeline, se necessário para a ação personalizada. Esse comando também retornará quaisquer valores secretos definidos para a ação, se houver algum definido.

Comando:

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

Saída:

```
{
  "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/bPxrRfiCYEXAMPLEKEY",
    "sessionToken":
    "fICCQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZncvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+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"
}
}

```


- Para API obter detalhes, consulte [GetJobDetails](#) na Referência de AWS CLI Comandos.

get-pipeline-state

O código de exemplo a seguir mostra como usar `get-pipeline-state`.

AWS CLI

Para obter informações sobre o estado de um gasoduto

Este exemplo retorna o estado mais recente de um pipeline chamado `MyFirstPipeline`.

Comando:

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

Saída:

```
{
  "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
}

```

- Para API obter detalhes, consulte [GetPipelineState](#) na Referência de AWS CLI Comandos.

get-pipeline

O código de exemplo a seguir mostra como usar `get-pipeline`.

AWS CLI

Para visualizar a estrutura de um pipeline

Este exemplo retorna a estrutura de um pipeline chamado `MyFirstPipeline`.

Comando:

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

Saída:

```

{
  "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
}
}

```

- Para API obter detalhes, consulte [GetPipeline](#) na Referência de AWS CLI Comandos.

list-action-executions

O código de exemplo a seguir mostra como usar `list-action-executions`.

AWS CLI

Para listar as execuções de ações

O `list-action-executions` exemplo a seguir mostra os detalhes da execução da ação de um pipeline, como ID de execução da ação, artefatos de entrada, artefatos de saída, resultado da execução e status.

```
aws codepipeline list-action-executions \
  --pipeline-name myPipeline
```

Saída:

```
{
  "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-111111720d3",
  "actionExecutionId": "EXAMPLE5-abb4-4192-9031-1111113a7b0",
  "pipelineVersion": 12,
  "stageName": "Source",
```

```
"actionName": "Source",
"startTime": 1598572624.387,
"lastUpdateTime": 1598572628.16,
"status": "Succeeded",
"input": {
  "actionTypeId": {
    "categoryId": "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":
"1111111ad99dcd35914c00b7fbea13995EXAMPLE",
    "externalExecutionSummary": "Edited template.yml",
    "externalExecutionUrl": "https://myaddress.com"
  },
  "outputVariables": {
    "AuthorDate": "2020-05-08T17:45:43Z",
    "BranchName": "production",
```

```
        "CommitId": "EXAMPLEad99dcd35914c00b7fbea139951111111",
        "CommitMessage": "Edited template.yml",
        "CommitterDate": "2020-05-08T17:45:43Z",
        "RepositoryName": "my-repo"
      }
    },
    . . . .
```

Para obter mais informações, consulte [Exibir execuções de ações \(CLI\)](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [ListActionExecutions](#) na Referência de AWS CLI Comandos.

list-action-types

O código de exemplo a seguir mostra como usar `list-action-types`.

AWS CLI

Para ver os tipos de ação disponíveis

Usado sozinho, o `list-action-types` comando retorna a estrutura de todas as ações disponíveis para sua AWS conta. Este exemplo usa a `action-owner-filter` opção `--` para retornar somente ações personalizadas.

Comando:

```
aws codepipeline list-action-types --action-owner-filter Custom
```

Saída:

```
{
  "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}"
    }
  }
]
}

```

- Para API obter detalhes, consulte [ListActionTypes](#) na Referência de AWS CLI Comandos.

list-pipeline-executions

O código de exemplo a seguir mostra como usar `list-pipeline-executions`.

AWS CLI

Para ver o histórico de execução do pipeline

O `list-pipeline-executions` exemplo a seguir mostra o histórico de execução de um funil em sua AWS conta.

```

aws codepipeline list-pipeline-executions \
  --pipeline-name MyPipeline

```

Saída:

```

{
  "pipelineExecutionSummaries": [
    {
      "lastUpdateTime": 1496380678.648,
      "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
      "startTime": 1496380258.243,
      "status": "Succeeded"
    },
    {
      "lastUpdateTime": 1496591045.634,
      "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d841-d7eu3EXAMPLE",
      "startTime": 1496590401.222,
      "status": "Succeeded"
    },
  ],
}

```

```
    {
      "lastUpdateTime": 1496946071.6456,
      "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
      "startTime": 1496945471.5645,
      "status": "Succeeded"
    }
  ]
}
```

Para obter mais informações, consulte [Exibir histórico de execução](#) no Guia AWS CodePipeline do usuário.

- Para API obter detalhes, consulte [ListPipelineExecutions](#) na Referência de AWS CLI Comandos.

list-pipelines

O código de exemplo a seguir mostra como usar `list-pipelines`.

AWS CLI

Para ver uma lista de pipelines

Este exemplo lista todos os AWS CodePipeline pipelines associados à AWS conta do usuário.

Comando:

```
aws codepipeline list-pipelines
```

Saída:

```
{
  "pipelines": [
    {
      "updated": 1439504274.641,
      "version": 1,
      "name": "MyFirstPipeline",
      "created": 1439504274.641
    },
    {
      "updated": 1436461837.992,
      "version": 2,
      "name": "MySecondPipeline",
```

```
        "created": 1436460801.381
      }
    ]
  }
```

- Para API obter detalhes, consulte [ListPipelines](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags

O `list-tags-for-resource` exemplo a seguir recupera uma lista de todas as tags anexadas ao recurso de pipeline especificado.

```
aws codepipeline list-tags-for-resource \
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

Saída:

```
{
  "tags": {
    "Project": "ProjectA",
    "IscontainerBased": "true"
  }
}
```

Para obter mais informações, consulte [Exibir tags de um pipeline \(CLI\)](#) no Guia AWS CodePipeline do usuário.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-webhooks

O código de exemplo a seguir mostra como usar `list-webhooks`.

AWS CLI

Para listar webhooks

O `list-webhooks` exemplo a seguir recupera uma lista de todas as tags anexadas ao recurso de pipeline especificado.

```
aws codepipeline list-webhooks \
  --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \
  --region "eu-central-1"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Listar webhooks em sua conta](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [ListWebhooks](#) na Referência de AWS CLI Comandos.

poll-for-jobs

O código de exemplo a seguir mostra como usar `poll-for-jobs`.

AWS CLI

Para ver todas as vagas disponíveis

Este exemplo retorna informações sobre quaisquer trabalhos nos quais um funcionário possa atuar. Este exemplo usa um JSON arquivo predefinido (`MyActionTypeInfo.json`) para fornecer informações sobre o tipo de ação para o qual o funcionário processa trabalhos. Esse comando é usado somente para ações personalizadas. Quando esse comando é chamado, AWS CodePipeline retorna credenciais temporárias para o bucket Amazon S3 usado para armazenar artefatos para o pipeline. Esse comando também retornará quaisquer valores secretos definidos para a ação, se houver algum definido.

Comando:

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

JSON conteúdo de amostra do arquivo:

```
{
  "actionTypeId": {
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
    "ProjectName": "MyJenkinsTestProject"
  }
}
```

Saída:

```
{
  "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":
    "fICCQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zvhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs
  },
  "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"
  }
]
}
```

- Para API obter detalhes, consulte [PollForJobs](#) na Referência de AWS CLI Comandos.

put-webhook

O código de exemplo a seguir mostra como usar `put-webhook`.

AWS CLI

Para criar um webhook

O `put-webhook` exemplo a seguir cria um webhook para uma ação de origem da GitHub versão 1. Depois de criar o webhook, você deve usar o comando `register-webhook-with-third-party` para registrá-lo.

```
aws codepipeline put-webhook \
  --cli-input-json file://webhook_json.json \
  --region "eu-central-1"
```

Conteúdo de `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"
    }
  }
}
```

```

    }
}

```

Saída:

```

{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Criar um webhook para uma GitHub fonte](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [PutWebhook](#) na Referência de AWS CLI Comandos.

retry-stage-execution

O código de exemplo a seguir mostra como usar `retry-stage-execution`.

AWS CLI

Para tentar novamente uma ação com falha

O `retry-stage-execution` exemplo a seguir repete um estágio com falha na ação.

```
aws codepipeline retry-stage-execution \  
  --pipeline-name MyPipeline \  
  --stage-name Deploy \  
  --pipeline-execution-id b59babff-5f34-EXAMPLE \  
  --retry-mode FAILED_ACTIONS
```

Saída:

```
{  
  "pipelineExecutionId": "b59babff-5f34-EXAMPLE"  
}
```

Para obter mais informações, consulte [Repetir ações com falha \(CLI\)](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [RetryStageExecution](#) na Referência de AWS CLI Comandos.

start-pipeline-execution

O código de exemplo a seguir mostra como usar `start-pipeline-execution`.

AWS CLI

Para executar a revisão mais recente por meio de um pipeline

Este exemplo executa a revisão mais recente presente no estágio de origem de um pipeline por meio do pipeline chamado "MyFirstPipeline".

Comando:

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

Saída:

```
{
```

```
"pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"  
}
```

- Para API obter detalhes, consulte [StartPipelineExecution](#) na Referência de AWS CLI Comandos.

stop-pipeline-execution

O código de exemplo a seguir mostra como usar stop-pipeline-execution.

AWS CLI

Para interromper a execução de um pipeline

O stop-pipeline-execution exemplo a seguir usa como padrão esperar até que as ações em andamento sejam concluídas e, em seguida, interrompa o pipeline. Não é possível optar por interromper e aguardar se a execução já estiver em um estado Stopping (Interrompendo). É possível optar por interromper e abandonar uma execução que já está em um estado Stopping (Interrompendo).

```
aws codepipeline stop-pipeline-execution \  
  --pipeline-name MyFirstPipeline \  
  --pipeline-execution-id d-EXAMPLE \  
  --reason "Stopping pipeline after the build action is done"
```

Esse comando não retorna uma saída.

Para obter mais informações, consulte [Interromper a execução de um pipeline \(CLI\)](#) no Guia AWS CodePipeline do usuário.

- Para API obter detalhes, consulte [StopPipelineExecution](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir associa um conjunto de tags fornecidas a um pipeline. Use esse comando para adicionar ou editar tags.

```
aws codepipeline tag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar tags a um pipeline \(CLI\)](#) no Guia do AWS CodePipeline usuário.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover AWS tags de um recurso de conexões

O `untag-resource` exemplo a seguir remove uma tag do recurso especificado.

```
aws codepipeline untag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tag-keys Project IscontainerBased
```

Este comando não produz saída.

Para obter mais informações, consulte [Remover tags de um pipeline \(CLI\)](#) no Guia AWS CodePipeline do usuário.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-pipeline

O código de exemplo a seguir mostra como usar `update-pipeline`.

AWS CLI

Para atualizar a estrutura de um pipeline

Este exemplo usa o comando `update-pipeline` com o argumento `--cli-input-json`. Este exemplo usa um JSON arquivo predefinido (`MyFirstPipeline.json`) para atualizar a estrutura de um pipeline. AWS CodePipeline reconhece o nome do pipeline contido no JSON arquivo e, em seguida, aplica todas as alterações dos campos modificados na estrutura do pipeline para atualizar o pipeline.

Use as diretrizes a seguir ao criar o JSON arquivo predefinido:

Se você estiver trabalhando com uma estrutura de pipeline recuperada usando o comando `get-pipeline`, deverá remover a seção de metadados da estrutura de pipeline no JSON arquivo (as linhas `"metadados": {}` e os campos `"criado"`, `"pipelineARN"` e `"atualizado"`). O nome do pipeline não pode ser alterado.

Comando:

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

Conteúdo JSON do arquivo de amostra:

```
{
  "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
}
}
```

Saída:

```
{
```

```
"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
}
```

- Para API obter detalhes, consulte [UpdatePipeline](#) na Referência de AWS CLI Comandos.

AWS CodeStar exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS CodeStar.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-team-member

O código de exemplo a seguir mostra como usar `associate-team-member`.

AWS CLI

Para adicionar um membro da equipe a um projeto

O `associate-team-member` exemplo a seguir torna o `intern` usuário um visualizador no projeto com o ID especificado.

```
aws codestar associate-team-member \  
  --project-id my-project \  
  --user-arn arn:aws:iam::123456789012:user/intern \  
  --project-role Viewer
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AssociateTeamMember](#) na Referência de AWS CLI Comandos.

create-project

O código de exemplo a seguir mostra como usar `create-project`.

AWS CLI

Para criar um projeto

O `create-project` exemplo a seguir usa um arquivo JSON de entrada para criar um CodeStar projeto.

```
aws codestar create-project \  
  --cli-input-json file://create-project.json
```

Conteúdo de `create-project.json`:

```
{  
  "name": "Custom Project",
```



```
"id": "custom-project",
"sourceCode": [
  {
    "source": {
      "s3": {
        "bucketName": "codestar-artifacts",
        "bucketKey": "nodejs-function.zip"
      }
    },
    "destination": {
      "codeCommit": {
        "name": "codestar-custom-project"
      }
    }
  }
],
"toolchain": {
  "source": {
    "s3": {
      "bucketName": "codestar-artifacts",
      "bucketKey": "toolchain.yml"
    }
  },
  "roleArn": "arn:aws:iam::123456789012:role/service-role/aws-codestar-
service-role",
  "stackParameters": {
    "ProjectId": "custom-project"
  }
}
}
```

Saída:

```
{
  "id": "my-project",
  "arn": "arn:aws:codestar:us-east-2:123456789012:project/custom-project"
}
```

Para ver um tutorial que inclui exemplos de código e modelos para um projeto personalizado, consulte Criar um projeto AWS CodeStar com o AWS CLI < <https://docs.aws.amazon.com/codestar/latest/userguide/cli-tutorial.html>> no Guia do usuário.AWS CodeStar

- Para API obter detalhes, consulte [CreateProject](#)na Referência de AWS CLI Comandos.

create-user-profile

O código de exemplo a seguir mostra como usar `create-user-profile`.

AWS CLI

Para criar um perfil de usuário

O `create-user-profile` exemplo a seguir cria um perfil de usuário para o IAM usuário com o especificadoARN.

```
aws codestar create-user-profile \  
  --user-arn arn:aws:iam::123456789012:user/intern \  
  --display-name Intern \  
  --email-address intern@example.com
```

Saída:

```
{  
  "userArn": "arn:aws:iam::123456789012:user/intern",  
  "displayName": "Intern",  
  "emailAddress": "intern@example.com",  
  "sshPublicKey": "",  
  "createdTimestamp": 1572552308.607,  
  "lastModifiedTimestamp": 1572552308.607  
}
```

- Para API obter detalhes, consulte [CreateUserProfile](#) na Referência de AWS CLI Comandos.

delete-project

O código de exemplo a seguir mostra como usar `delete-project`.

AWS CLI

Para excluir um projeto

O `delete-project` exemplo a seguir exclui o projeto especificado.

```
aws codestar delete-project \  
  --project-id my-project
```

Saída:

```
{
  "projectArn": "arn:aws:codestar:us-east-2:123456789012:project/my-project"
}
```

- Para API obter detalhes, consulte [DeleteProject](#) na Referência de AWS CLI Comandos.

delete-user-profile

O código de exemplo a seguir mostra como usar delete-user-profile.

AWS CLI

Para excluir um perfil de usuário

O delete-user-profile exemplo a seguir exclui o perfil do usuário com o especificado ARN.

```
aws codestar delete-user-profile \
  --user-arn arn:aws:iam::123456789012:user/intern
```

Saída:

```
{
  "userArn": "arn:aws:iam::123456789012:user/intern"
}
```

- Para API obter detalhes, consulte [DeleteUserProfile](#) na Referência de AWS CLI Comandos.

describe-project

O código de exemplo a seguir mostra como usar describe-project.

AWS CLI

Para ver um projeto

O describe-project exemplo a seguir recupera detalhes sobre o projeto especificado.

```
aws codestar describe-project \
  --id my-project
```

Saída:

```
{
  "name": "my project",
  "id": "my-project",
  "arn": "arn:aws:codestar:us-west-2:123456789012:project/my-project",
  "description": "My first CodeStar project.",
  "createdTimeStamp": 1572547510.128,
  "status": {
    "state": "CreateComplete"
  }
}
```

- Para API obter detalhes, consulte [DescribeProject](#) na Referência de AWS CLI Comandos.

describe-user-profile

O código de exemplo a seguir mostra como usar `describe-user-profile`.

AWS CLI

Para ver um perfil de usuário

O `describe-user-profile` exemplo a seguir recupera detalhes sobre o perfil do usuário com o especificado ARN.

```
aws codestar describe-user-profile \
  --user-arn arn:aws:iam::123456789012:user/intern
```

Saída:

```
{
  "userArn": "arn:aws:iam::123456789012:user/intern",
  "displayName": "Intern",
  "emailAddress": "intern@example.com",
  "sshPublicKey": "intern",
  "createdTimestamp": 1572552308.607,
  "lastModifiedTimestamp": 1572553495.47
}
```

- Para API obter detalhes, consulte [DescribeUserProfile](#) na Referência de AWS CLI Comandos.

disassociate-team-member

O código de exemplo a seguir mostra como usar `disassociate-team-member`.

AWS CLI

Para remover um membro da equipe

O `disassociate-team-member` exemplo a seguir remove o usuário com o especificado ARN do projeto `my-project`.

```
aws codestar disassociate-team-member \  
  --project-id my-project \  
  --user-arn arn:aws:iam::123456789012:user/intern
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DisassociateTeamMember](#) na Referência de AWS CLI Comandos.

list-projects

O código de exemplo a seguir mostra como usar `list-projects`.

AWS CLI

Para ver projetos

O `list-projects` exemplo a seguir recupera uma lista de projetos na região atual.

```
aws codestar list-projects
```

Saída:

```
{  
  "projects": [  
    {  
      "projectId": "intern-projects",  
      "projectArn": "arn:aws:codestar:us-west-2:123456789012:project/intern-  
projects"  
    },  
    {  
      "projectId": "my-project",
```

```

        "projectArn": "arn:aws:codestar:us-west-2:123456789012:project/my-
project"
      }
    ]
  }

```

- Para API obter detalhes, consulte [ListProjects](#) na Referência de AWS CLI Comandos.

list-resources

O código de exemplo a seguir mostra como usar `list-resources`.

AWS CLI

Para ver os recursos

O `list-resources` exemplo a seguir recupera uma lista de recursos para o projeto especificado.

```

aws codestar list-resources \
  --id my-project

```

Saída:

```

{
  "resources": [
    {
      "id": "arn:aws:execute-api:us-east-2:123456789012:r3wxmplbv8"
    },
    {
      "id": "arn:aws:codedeploy:us-east-2:123456789012:application:awscodestar-my-project-lambda-ServerlessDeploymentApplication-PF0LXMPL1KA0"
    },
    {
      "id": "arn:aws:s3:::aws-codestar-us-east-2-123456789012-my-project-pipe"
    },
    {
      "id": "arn:aws:lambda:us-east-2:123456789012:function:awscodestar-my-project-lambda-GetHelloWorld-16W3LVXMPLNNS"
    },
    {

```

```
    "id": "arn:aws:cloudformation:us-east-2:123456789012:stack/awscodestar-
my-project-lambda/b4904ea0-fc20-xmpl-bec6-029123b1cc42"
  },
  {
    "id": "arn:aws:cloudformation:us-east-2:123456789012:stack/awscodestar-
my-project/1b133f30-fc20-xmpl-a93a-0688c4290cb8"
  },
  {
    "id": "arn:aws:iam::123456789012:role/CodeStarWorker-my-project-
ToolChain"
  },
  {
    "id": "arn:aws:iam::123456789012:policy/CodeStar_my-
project_PermissionsBoundary"
  },
  {
    "id": "arn:aws:s3::aws-codestar-us-east-2-123456789012-my-project-app"
  },
  {
    "id": "arn:aws:codepipeline:us-east-2:123456789012:my-project-Pipeline"
  },
  {
    "id": "arn:aws:codedeploy:us-east-2:123456789012:deploymentgroup:my-
project/awscodestar-my-project-lambda-GetHelloWorldDeploymentGroup-P7YWXMPLT0QB"
  },
  {
    "id": "arn:aws:iam::123456789012:role/CodeStar-my-project-Execution"
  },
  {
    "id": "arn:aws:iam::123456789012:role/CodeStarWorker-my-project-
CodeDeploy"
  },
  {
    "id": "arn:aws:codebuild:us-east-2:123456789012:project/my-project"
  },
  {
    "id": "arn:aws:iam::123456789012:role/CodeStarWorker-my-project-
CloudFormation"
  },
  {
    "id": "arn:aws:codecommit:us-east-2:123456789012:Go-project"
  }
]
```

```
}
```

- Para API obter detalhes, consulte [ListResources](#) na Referência de AWS CLI Comandos.

list-tags-for-project

O código de exemplo a seguir mostra como usar `list-tags-for-project`.

AWS CLI

Para ver as tags de um projeto

O `list-tags-for-project` exemplo a seguir recupera as tags anexadas ao projeto especificado.

```
aws codestar list-tags-for-project \  
  --id my-project
```

Saída:

```
{  
  "tags": {  
    "Department": "Marketing",  
    "Team": "Website"  
  }  
}
```

- Para API obter detalhes, consulte [ListTagsForProject](#) na Referência de AWS CLI Comandos.

list-team-members

O código de exemplo a seguir mostra como usar `list-team-members`.

AWS CLI

Para ver uma lista de membros da equipe

O `list-team-members` exemplo a seguir recupera uma lista de usuários associados ao projeto especificado.

```
aws codestar list-team-members \  
  --id my-project
```



```
--project-id my-project
```

Saída:

```
{
  "teamMembers": [
    {
      "userArn": "arn:aws:iam::123456789012:user/admin",
      "projectRole": "Owner",
      "remoteAccessAllowed": false
    },
    {
      "userArn": "arn:aws:iam::123456789012:user/intern",
      "projectRole": "Contributor",
      "remoteAccessAllowed": false
    }
  ]
}
```

- Para API obter detalhes, consulte [ListTeamMembers](#) na Referência de AWS CLI Comandos.

list-user-profiles

O código de exemplo a seguir mostra como usar `list-user-profiles`.

AWS CLI

Para ver uma lista de perfis de usuário

O `list-user-profiles` exemplo a seguir recupera uma lista de todos os perfis de usuário na região atual.

```
aws codestar list-user-profiles
```

Saída:

```
{
  "userProfiles": [
    {
      "userArn": "arn:aws:iam::123456789012:user/admin",
      "displayName": "me",
      "emailAddress": "me@example.com",
```

```
        "sshPublicKey": ""
      },
      {
        "userArn": "arn:aws:iam::123456789012:user/intern",
        "displayName": "Intern",
        "emailAddress": "intern@example.com",
        "sshPublicKey": "intern"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [ListUserProfiles](#) na Referência de AWS CLI Comandos.

tag-project

O código de exemplo a seguir mostra como usar `tag-project`.

AWS CLI

Para anexar uma tag a um projeto

O `tag-project` exemplo a seguir adiciona uma tag chamada `Department` e um valor de `Marketing` ao projeto especificado.

```
aws codestar tag-project \
  --id my-project \
  --tags Department=Marketing
```

Saída:

```
{
  "tags": {
    "Department": "Marketing"
  }
}
```

- Para API obter detalhes, consulte [TagProject](#) na Referência de AWS CLI Comandos.

untag-project

O código de exemplo a seguir mostra como usar `untag-project`.

AWS CLI

Para remover uma tag de um projeto

O `untag-project` exemplo a seguir remove qualquer tag com um nome de chave `Team` de do projeto específico.

```
aws codestar untag-project \  
  --id my-project \  
  --tags Team
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagProject](#) na Referência de AWS CLI Comandos.

update-project

O código de exemplo a seguir mostra como usar `update-project`.

AWS CLI

Para atualizar um projeto

O `update-project` exemplo a seguir adiciona uma descrição ao projeto especificado.

```
aws codestar update-project \  
  --id my-project \  
  --description "My first CodeStar project"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateProject](#) na Referência de AWS CLI Comandos.

update-team-member

O código de exemplo a seguir mostra como usar `update-team-member`.

AWS CLI

Para modificar um membro da equipe

O `update-team-member` exemplo a seguir torna o usuário especificado um colaborador em um projeto e concede a ele acesso remoto aos recursos do projeto.

```
aws codestar update-team-member \  
  --project-id my-project \  
  --user-arn arn:aws:iam::123456789012:user/intern \  
  --project-role Contributor -\  
  --remote-access-allowed
```

Saída:

```
{  
  "userArn": "arn:aws:iam::123456789012:user/intern",  
  "projectRole": "Contributor",  
  "remoteAccessAllowed": true  
}
```

- Para API obter detalhes, consulte [UpdateTeamMember](#) na Referência de AWS CLI Comandos.

update-user-profile

O código de exemplo a seguir mostra como usar `update-user-profile`.

AWS CLI

Para modificar um perfil de usuário

O `update-user-profile` exemplo a seguir adiciona a SSH chave especificada ao usuário especificado.

```
aws codestar update-user-profile \  
  --ssh-public-key intern \  
  --user-arn arn:aws:iam::123456789012:user/intern
```

Saída:

```
{  
  "userArn": "arn:aws:iam::123456789012:user/intern",  
  "displayName": "Intern",  
  "emailAddress": "intern@example.com",  
  "sshPublicKey": "intern",  
  "createdTimestamp": 1572552308.607,  
  "lastModifiedTimestamp": 1572553495.47  
}
```

```
}
```

- Para API obter detalhes, consulte [UpdateUserProfile](#) na Referência de AWS CLI Comandos.

AWS CodeStar Exemplos de notificações usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com AWS CodeStar notificações.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-notification-rule

O código de exemplo a seguir mostra como usar `create-notification-rule`.

AWS CLI

Para criar uma regra de notificação

O `create-notification-rule` exemplo a seguir usa um JSON arquivo chamado `rule.json` para criar uma regra de notificação com o nome `MyNotificationRule` de um repositório nomeado `MyDemoRepo` na AWS conta especificada. As notificações com o tipo de `FULL` detalhe são enviadas para o SNS tópico de destino especificado da Amazon quando filiais e tags são criadas.

```
aws codestar-notifications create-notification-rule \  
  --cli-input-json file://rule.json
```

Conteúdo de `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"
}
```

Saída:

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
}
```

Para obter mais informações, consulte [Criar uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [CreateNotificationRule](#) na Referência de AWS CLI Comandos.

delete-notification-rule

O código de exemplo a seguir mostra como usar `delete-notification-rule`.

AWS CLI

Para excluir uma regra de notificação

O `delete-notification-rule` exemplo a seguir exclui a regra de notificação especificada.

```
aws codestar-notifications delete-notification-rule \
```

```
--arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

Saída:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Para obter mais informações, consulte [Excluir uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [DeleteNotificationRule](#) na Referência de AWS CLI Comandos.

delete-target

O código de exemplo a seguir mostra como usar delete-target.

AWS CLI

Para excluir um alvo de regra de notificação

O delete-target exemplo a seguir remove o alvo especificado de todas as regras de notificação configuradas para usá-lo como alvo e, em seguida, exclui o alvo.

```
aws codestar-notifications delete-target \  
  --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \  
  --force-unsubscribe-all
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um alvo de regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [DeleteTarget](#) na Referência de AWS CLI Comandos.

describe-notification-rule

O código de exemplo a seguir mostra como usar describe-notification-rule.

AWS CLI

Para recuperar detalhes de uma regra de notificação

O `describe-notification-rule` exemplo a seguir recupera os detalhes da regra de notificação especificada.

```
aws codestar-notifications describe-notification-rule \  
  --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Exibir regras de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [DescribeNotificationRule](#) na Referência de AWS CLI Comandos.

list-event-types

O código de exemplo a seguir mostra como usar `list-event-types`.

AWS CLI

Para obter uma lista de tipos de eventos para uma regra de notificação

O `list-event-types` exemplo a seguir recupera uma lista filtrada de todos os tipos de eventos de notificação disponíveis para CodeDeploy aplicativos. Se, em vez disso, você não usar nenhum filtro, o comando retornará todos os tipos de eventos de notificação para todos os tipos de recursos.

```
aws codestar-notifications list-event-types \  
  --filters Name=SERVICE_NAME,Value=CodeDeploy
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

Para obter mais informações, consulte [Criar uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [ListEventTypes](#) na Referência de AWS CLI Comandos.

list-notification-rules

O código de exemplo a seguir mostra como usar `list-notification-rules`.

AWS CLI

Para recuperar uma lista de regras de notificação

O `list-notification-rules` exemplo a seguir recupera uma lista de todas as regras de notificação na AWS região especificada.

```
aws codestar-notifications list-notification-rules --region us-east-1
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Exibir regras de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [ListNotificationRules](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para obter uma lista de tags anexadas a uma regra de notificação

O `list-tags-for-resource` exemplo a seguir recupera uma lista de todas as tags anexadas à regra de notificação especificada. Neste exemplo, a regra de notificação atualmente não tem tags associadas a ela.

```
aws codestar-notifications list-tags-for-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE
```

Saída:

```
{  
  "Tags": {}  
}
```

Para obter mais informações, consulte [Criar uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-targets

O código de exemplo a seguir mostra como usar `list-targets`.

AWS CLI

Para recuperar uma lista de alvos de regras de notificação

O `list-targets` exemplo a seguir recupera uma lista de todos os alvos da regra de notificação na AWS região especificada.

```
aws codestar-notifications list-targets \  
  --region us-east-1
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Exibir metas de regras de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [ListTargets](#) na Referência de AWS CLI Comandos.

subscribe

O código de exemplo a seguir mostra como usar `subscribe`.

AWS CLI

Para adicionar um alvo a uma regra de notificação

O `subscribe` exemplo a seguir adiciona um SNS tópico da Amazon como destino para a regra de notificação especificada.

```
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
```

Saída:

```
{
```

```
"Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Para obter mais informações, consulte [Adicionar ou remover um SNS tópico da Amazon como alvo para uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [Inscrever-se](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a uma regra de notificação

O `tag-resource` exemplo a seguir adiciona uma tag com o nome da chave `Team` e o valor de `Li_Juan` à regra de notificação especificada.

```
aws codestar-notifications tag-resource \
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE \
  --tags Team=Li_Juan
```

Saída:

```
{
  "Tags": {
    "Team": "Li_Juan"
  }
}
```

Para obter mais informações, consulte [Criar uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

unsubscribe

O código de exemplo a seguir mostra como usar `unsubscribe`.

AWS CLI

Para remover um alvo de uma regra de notificação

O `unsubscribe` exemplo a seguir remove um SNS tópico da Amazon como alvo da regra de notificação especificada.

```
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
```

Saída:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
  "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"  
}
```

Para obter mais informações, consulte [Adicionar ou remover um SNS tópico da Amazon como alvo para uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [Cancelar inscrição](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de uma regra de notificação

O `untag-resource` exemplo a seguir remove a tag com o nome `Team` da chave da regra de notificação especificada.

```
aws codestar-notifications untag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tag-key Team
```

```
--tag-keys Team
```

Este comando não produz saída.

Para obter mais informações, consulte [Editar uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-notification-rule

O código de exemplo a seguir mostra como usar update-notification-rule.

AWS CLI

Para atualizar uma regra de notificação

O update-notification-rule exemplo a seguir atualiza uma regra de notificação nomeada MyNotificationRule na AWS conta 123456789012 usando um JSON arquivo chamado update.json.

```
aws codestar-notifications update-notification-rule \  
--cli-input-json file://update.json
```

Conteúdo de update.json:

```
{  
  "Name": "MyUpdatedNotificationRule",  
  "EventIds": [  
    "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"  
}
```

Saída:

```
{
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

Para obter mais informações, consulte [Editar uma regra de notificação](#) no Guia do usuário do AWS Developer Tools Console.

- Para API obter detalhes, consulte [UpdateNotificationRule](#) na Referência de AWS CLI Comandos.

CodeConnections exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with CodeConnections.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-connection

O código de exemplo a seguir mostra como usar `create-connection`.

AWS CLI

Para criar uma conexão

O `create-connection` exemplo a seguir mostra como criar uma conexão com um repositório de terceiros. Este exemplo cria uma conexão em que o provedor terceirizado é o Bitbucket.

Uma conexão criada por meio do AWS CLI ou AWS CloudFormation está no status Pendente por padrão. Depois de criar uma conexão com o CLI ou AWS CloudFormation, use o console para editar a conexão e tornar seu status Disponível.

```
aws codestar-connections create-connection \  
  --provider-type Bitbucket \  
  --connection-name MyConnection
```

Saída:

```
{  
  "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"  
}
```

Para obter mais informações, consulte [Criar uma conexão](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [CreateConnection](#) na Referência de AWS CLI Comandos.

create-host

O código de exemplo a seguir mostra como usar create-host.

AWS CLI

Para criar um host

O create-host exemplo a seguir mostra como criar um host para representar o endpoint da infraestrutura em que seu provedor terceirizado está instalado. Este exemplo cria um host em que o provedor instalado de terceiros é o GitHub Enterprise Server.

Um host criado por meio do AWS CLI está no status Pendente por padrão. Depois de criar um host com o CLI, use o console ou o CLI para configurar o host para tornar seu status Disponível.

```
aws codestar-connections create-host \  
  --name MyHost \  
  --provider-type GitHubEnterpriseServer \  
  --provider-endpoint "https://my-instance.dev"
```

Saída:

```
{
  "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605"
}
```

Para obter mais informações, consulte [Criar um host \(CLI\)](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [CreateHost](#) na Referência de AWS CLI Comandos.

delete-connection

O código de exemplo a seguir mostra como usar `delete-connection`.

AWS CLI

Para excluir uma conexão

O `delete-connection` exemplo a seguir mostra como excluir uma conexão.

```
aws codestar-connections delete-connection \
  --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir uma conexão \(CLI\)](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [DeleteConnection](#) na Referência de AWS CLI Comandos.

delete-host

O código de exemplo a seguir mostra como usar `delete-host`.

AWS CLI

Para excluir um host

O `delete-host` exemplo a seguir mostra como excluir um host. Para poder excluir um host, primeiro é necessário excluir todas as conexões associadas a ele.

```
aws codestar-connections delete-host \  
  --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-  
Host-28aef605"
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um host \(CLI\)](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [DeleteHost](#) na Referência de AWS CLI Comandos.

get-connection

O código de exemplo a seguir mostra como usar `get-connection`.

AWS CLI

Para obter informações sobre uma conexão

O `get-connection` exemplo a seguir mostra detalhes sobre uma conexão.

```
aws codestar-connections get-connection \  
  --connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Saída:

```
{  
  "Connection": {  
    "ConnectionName": "MyConnection",  
    "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",  
    "ProviderType": "Bitbucket",  
    "OwnerAccountId": "123456789012",  
    "ConnectionStatus": "AVAILABLE"  
  }  
}
```

Para obter mais informações, consulte [Exibir detalhes da conexão](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [GetConnection](#) na Referência de AWS CLI Comandos.

get-host

O código de exemplo a seguir mostra como usar `get-host`.

AWS CLI

Para obter informações sobre um anfitrião

O `get-host` exemplo a seguir mostra detalhes sobre um host:

```
aws codestar-connections get-host \  
  --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/  
MyHost-28aef605
```

Saída:

```
{  
  "Name": "MyHost",  
  "Status": "AVAILABLE",  
  "ProviderType": "GitHubEnterpriseServer",  
  "ProviderEndpoint": "https://test-instance-1.dev/"  
}
```

Para obter mais informações, consulte [Exibir detalhes do host \(CLI\)](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [GetHost](#) na Referência de AWS CLI Comandos.

list-connections

O código de exemplo a seguir mostra como usar `list-connections`.

AWS CLI

Para listar conexões

O `list-connections` exemplo a seguir recupera uma lista de todas as conexões em sua conta para o tipo de provedor Bitbucket. :

```
aws codestar-connections list-connections \  
  --provider-type Bitbucket \  
  --max-results 5 \  
  --
```

```
--next-token: next-token
```

Saída:

```
{
  "Connections": [
    {
      "ConnectionName": "my-connection",
      "ProviderType": "Bitbucket",
      "Status": "PENDING",
      "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
      "OwnerId": "123456789012"
    },
    {
      "ConnectionName": "my-other-connection",
      "ProviderType": "Bitbucket",
      "Status": "AVAILABLE",
      "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
      "OwnerId": "123456789012"
    }
  ],
  "NextToken": "next-token"
}
```

Para obter mais informações, consulte [Listar conexões \(CLI\)](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [ListConnections](#) na Referência de AWS CLI Comandos.

list-hosts

O código de exemplo a seguir mostra como usar `list-hosts`.

AWS CLI

Para listar anfitriões

O `list-hosts` exemplo a seguir recupera uma lista de todos os hosts em sua conta.

```
aws codestar-connections list-hosts
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Listar hosts \(CLI\)](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [ListHosts](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags

O `list-tags-for-resource` exemplo a seguir recupera uma lista de todas as tags anexadas ao recurso de conexões especificado.

```
aws codestar-connections list-tags-for-resource \
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Saída:

```
{
  "Tags": [
    {
      "Key": "Project",
      "Value": "ProjectA"
    }
  ]
}
```

```
    },  
    {  
      "Key": "ReadOnly",  
      "Value": "true"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Exibir tags para um recurso de conexões](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir associa um conjunto de tags fornecidas a uma conexão. Use esse comando para adicionar ou editar tags.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar tags a um recurso de conexões](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover AWS tags de um recurso de conexões

O seguinte `untag-resource` remove uma tag do recurso especificado.

```
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
```

Saída:

```
{  
  "Tags": []  
}
```

Para obter mais informações, consulte [Remover tags de um recurso de conexões](#) no Guia do usuário do console Developer Tools.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

Exemplos de identidade do Amazon Cognito usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface Amazon Cognito Identity.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-identity-pool

O código de exemplo a seguir mostra como usar `create-identity-pool`.

AWS CLI

Como criar um banco de identidades com o provedor de banco de identidades Cognito

Este exemplo cria um grupo de identidades chamado MyIdentityPool. Ele tem um provedor de banco de identidades Cognito. Identidades não autenticadas não são permitidas.

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="3n4b5urk1ft4f13mg5e62d9ado",ServerSideTokenCheck=false
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [CreateIdentityPool](#) na Referência de AWS CLI Comandos.

delete-identities

O código de exemplo a seguir mostra como usar delete-identities.

AWS CLI

Como excluir um banco de identidades

Este exemplo exclui um grupo de identidades.

Comando:

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-west-2:11111111-1111-1111-1111-111111111111"
```

Saída:

```
{
  "UnprocessedIdentityIds": []
}
```

- Para API obter detalhes, consulte [DeletIdentities](#) na Referência de AWS CLI Comandos.

delete-identity-pool

O código de exemplo a seguir mostra como usar delete-identity-pool.

AWS CLI

Como excluir um banco de identidades

O exemplo delete-identity-pool a seguir exclui o banco de identidades especificado.

Comando:

```
aws cognito-identity delete-identity-pool \
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteIdentityPool](#) na Referência de AWS CLI Comandos.

describe-identity-pool

O código de exemplo a seguir mostra como usar describe-identity-pool.

AWS CLI

Para descrever um pool de identidades

Este exemplo descreve um grupo de identidades.

Comando:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeIdentityPool](#) na Referência de AWS CLI Comandos.

get-identity-pool-roles

O código de exemplo a seguir mostra como usar `get-identity-pool-roles`.

AWS CLI

Para obter funções no pool de identidades

Este exemplo obtém funções do grupo de identidades.

Comando:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Saída:

```
{
  "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"
}
}
```

- Para API obter detalhes, consulte [GetIdentityPoolRoles](#) na Referência de AWS CLI Comandos.

list-identity-pools

O código de exemplo a seguir mostra como usar `list-identity-pools`.

AWS CLI

Como listar bancos de identidades

Este exemplo lista bancos de identidades. Há no máximo vinte identidades listadas.

Comando:

```
aws cognito-identity list-identity-pools --max-results 20
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListIdentityPools](#) na Referência de AWS CLI Comandos.

set-identity-pool-roles

O código de exemplo a seguir mostra como usar `set-identity-pool-roles`.

AWS CLI

Para definir funções do grupo de identidades

O `set-identity-pool-roles` exemplo a seguir define uma função do grupo de identidades.

```
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"
```

- Para API obter detalhes, consulte [SetIdentityPoolRoles](#) na Referência de AWS CLI Comandos.

update-identity-pool

O código de exemplo a seguir mostra como usar `update-identity-pool`.

AWS CLI

Para atualizar um grupo de identidades

Este exemplo atualiza um grupo de identidades. Ele define o nome como `MyIdentityPool`. Ele adiciona o Cognito como provedor de identidade. Ele não permite identidades não autenticadas.

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_11111111",ClientId="3n4b5urk1ft4f13mg5e62d9ado",ServerSideTokenCheck=false
```

Saída:

```
{  
  "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": "3n4b5urk1ft4f13mg5e62d9ado",  
    "ServerSideTokenCheck": false  
  }  
]  
}
```

- Para API obter detalhes, consulte [UpdateIdentityPool](#) na Referência de AWS CLI Comandos.

Exemplos de provedores de identidade do Amazon Cognito usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface Amazon Cognito Identity Provider.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-custom-attributes

O código de exemplo a seguir mostra como usar `add-custom-attributes`.

AWS CLI

Para adicionar um atributo personalizado

Este exemplo adiciona um atributo personalizado `CustomAttr 1` a um grupo de usuários. É do tipo `String` e requer no mínimo 1 caractere e no máximo 15. Não é obrigatório.

Comando:

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaaaa --custom-attributes Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- Para API obter detalhes, consulte [AddCustomAttributes](#) na Referência de AWS CLI Comandos.

admim-disable-user

O código de exemplo a seguir mostra como usar `admim-disable-user`.

AWS CLI

Para desativar um usuário

Este exemplo desativa o usuário `jane@example.com`.

Comando:

```
aws cognito-idp admin-disable-user --user-pool-id us-west-2_aaaaaaaaaa --username jane@example.com
```

- Para API obter detalhes, consulte [AdminDisableUser](#) na Referência de AWS CLI Comandos.

admim-enable-user

O código de exemplo a seguir mostra como usar `admim-enable-user`.

AWS CLI

Para habilitar um usuário

Este exemplo habilita o nome de usuário `jane@example.com`.

Comando:

```
aws cognito-idp admin-enable-user --user-pool-id us-west-2_aaaaaaaaaa --username jane@example.com
```

- Para API obter detalhes, consulte [AdminEnableUser](#) na Referência de AWS CLI Comandos.

admin-add-user-to-group

O código de exemplo a seguir mostra como usar `admin-add-user-to-group`.

AWS CLI

Para adicionar um usuário a um grupo

Este exemplo adiciona a usuária Jane ao grupo MyGroup.

Comando:

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --  
username Jane --group-name MyGroup
```

- Para API obter detalhes, consulte [AdminAddUserToGroup](#) na Referência de AWS CLI Comandos.

admin-confirm-sign-up

O código de exemplo a seguir mostra como usar `admin-confirm-sign-up`.

AWS CLI

Para confirmar o registro do usuário

Este exemplo confirma o usuário `jane@example.com`.

Comando:

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- Para API obter detalhes, consulte [AdminConfirmSignUp](#) na Referência de AWS CLI Comandos.

admin-create-user

O código de exemplo a seguir mostra como usar `admin-create-user`.

AWS CLI

Para criar um usuário

O `admin-create-user` exemplo a seguir cria um usuário com as configurações especificadas de endereço de e-mail e número de telefone.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [AdminCreateUser](#) na Referência de AWS CLI Comandos.

admin-delete-user-attributes

O código de exemplo a seguir mostra como usar `admin-delete-user-attributes`.

AWS CLI

Para excluir um atributo do usuário

Este exemplo exclui um atributo personalizado CustomAttr 1 para o usuário `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"
```

- Para API obter detalhes, consulte [AdminDeleteUserAttributes](#) na Referência de AWS CLI Comandos.

admin-delete-user

O código de exemplo a seguir mostra como usar `admin-delete-user`.

AWS CLI

Como excluir um usuário

Este exemplo exclui um usuário.

Comando:

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- Para API obter detalhes, consulte [AdminDeleteUser](#) na Referência de AWS CLI Comandos.

admin-forget-device

O código de exemplo a seguir mostra como usar `admin-forget-device`.

AWS CLI

Para esquecer um dispositivo

Este exemplo esquece o dispositivo para o nome de usuário `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
```

- Para API obter detalhes, consulte [AdminForgetDevice](#) na Referência de AWS CLI Comandos.

admin-get-device

O código de exemplo a seguir mostra como usar `admin-get-device`.

AWS CLI

Para obter um dispositivo

Este exemplo obtém um dispositivo para o nome de usuário `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
```

- Para API obter detalhes, consulte [AdminGetDevice](#) na Referência de AWS CLI Comandos.

admin-get-user

O código de exemplo a seguir mostra como usar `admin-get-user`.

AWS CLI

Como obter um usuário

Este exemplo obtém informações sobre o nome de usuário `jane@example.com`.

Comando:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

Saída:

```
{  
  "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
}
```

- Para API obter detalhes, consulte [AdminGetUser](#) na Referência de AWS CLI Comandos.

admin-initiate-auth

O código de exemplo a seguir mostra como usar admin-initiate-auth.

AWS CLI

Como iniciar a autorização

Este exemplo inicia a autorização usando o AUTH fluxo ADMIN_NO_SRP para o nome de usuário jane@example.com

O cliente deve ter o login API para autenticação baseada em servidor (ADMIN_NÃO_SRP) habilitado. AUTH

Use as informações da sessão no valor de retorno para chamar `admin-respond-to-auth - challenge`.

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
```

Saída:

```
{
  "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\"}"
  }
}
```

- Para API obter detalhes, consulte [AdminInitiateAuth](#) na Referência de AWS CLI Comandos.

admin-list-devices

O código de exemplo a seguir mostra como usar `admin-list-devices`.

AWS CLI

Para listar dispositivos para um usuário

Este exemplo lista dispositivos para o nome de usuário `jane@example.com`.

Comando:

```
aws cognito-idp admin-list-devices --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- Para API obter detalhes, consulte [AdminListDevices](#) na Referência de AWS CLI Comandos.

admin-list-groups-for-user

O código de exemplo a seguir mostra como usar `admin-list-groups-for-user`.

AWS CLI

Para listar grupos para um usuário

Este exemplo lista grupos para o nome de usuário `jane@example.com`.

Comando:

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [AdminListGroupForUser](#) na Referência de AWS CLI Comandos.

admin-list-user-auth-events

O código de exemplo a seguir mostra como usar `admin-list-user-auth-events`.

AWS CLI

Para listar eventos de autorização para um usuário

Este exemplo lista eventos de autorização para o nome de usuário `diego@example.com`.

Comando:

```
aws cognito-idp admin-list-user-auth-events --user-pool-id us-west-2_aaaaaaaa --  
username diego@example.com
```

- Para API obter detalhes, consulte [AdminListUserAuthEvents](#) na Referência de AWS CLI Comandos.

admin-remove-user-from-group

O código de exemplo a seguir mostra como usar `admin-remove-user-from-group`.

AWS CLI

Para remover um usuário de um grupo

Este exemplo remove `jane@example.com` de `SampleGroup`.

Comando:

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaa --  
username jane@example.com --group-name SampleGroup
```

- Para API obter detalhes, consulte [AdminRemoveUserFromGroup](#) na Referência de AWS CLI Comandos.

admin-reset-user-password

O código de exemplo a seguir mostra como usar `admin-reset-user-password`.

AWS CLI

Para redefinir uma senha de usuário

Este exemplo redefine a senha para `diego@example.com`.

Comando:

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- Para API obter detalhes, consulte [AdminResetUserPassword](#) na Referência de AWS CLI Comandos.

admin-set-user-mfa-preference

O código de exemplo a seguir mostra como usar `admin-set-user-mfa-preference`.

AWS CLI

Para definir a MFA preferência do usuário

Este exemplo define a SMS MFA preferência para o nome de usuário `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
```

- Para API obter detalhes, consulte [AdminSetUserMfaPreference](#) na Referência de AWS CLI Comandos.

admin-set-user-settings

O código de exemplo a seguir mostra como usar `admin-set-user-settings`.

AWS CLI

Para definir as configurações do usuário

Este exemplo define a preferência MFA de entrega do nome de usuário `diego@example.com` como EMAIL.

Comando:

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com --mfa-options DeliveryMedium=EMAIL
```


- Para API obter detalhes, consulte [AdminSetUserSettings](#) na Referência de AWS CLI Comandos.

admin-update-auth-event-feedback

O código de exemplo a seguir mostra como usar `admin-update-auth-event-feedback`.

AWS CLI

Para fornecer feedback sobre um evento de autorização

Este exemplo define o valor do feedback para um evento de autorização identificado pelo `event-id` como `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
```

- Para API obter detalhes, consulte [AdminUpdateAuthEventFeedback](#) na Referência de AWS CLI Comandos.

admin-update-device-status

O código de exemplo a seguir mostra como usar `admin-update-device-status`.

AWS CLI

Para atualizar o status do dispositivo

Este exemplo define o status do dispositivo lembrado para o dispositivo identificado pela chave do dispositivo como `not_remembered`.

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
```

- Para API obter detalhes, consulte [AdminUpdateDeviceStatus](#) na Referência de AWS CLI Comandos.

admin-update-user-attributes

O código de exemplo a seguir mostra como usar `admin-update-user-attributes`.

AWS CLI

Para atualizar os atributos do usuário

Este exemplo atualiza um atributo de usuário personalizado CustomAttr 1 para o usuário `diego@example.com`.

Comando:

```
aws cognito-idp admin-update-user-attributes --user-pool-id us-west-2_aaaaaaaa --username diego@example.com --user-attributes Name="custom:CustomAttr1",Value="Purple"
```

- Para API obter detalhes, consulte [AdminUpdateUserAttributes](#) na Referência de AWS CLI Comandos.

change-password

O código de exemplo a seguir mostra como usar `change-password`.

AWS CLI

Para alterar uma senha

Este exemplo altera uma senha.

Comando:

```
aws cognito-idp change-password --previous-password OldPassword --proposed-password NewPassword --access-token ACCESS_TOKEN
```

- Para API obter detalhes, consulte [ChangePassword](#) na Referência de AWS CLI Comandos.

confirm-forgot-password

O código de exemplo a seguir mostra como usar `confirm-forgot-password`.

AWS CLI

Para confirmar uma senha esquecida

Este exemplo confirma uma senha esquecida para o nome de usuário `diego@example.com`.

Comando:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --  
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Para API obter detalhes, consulte [ConfirmForgotPassword](#) na Referência de AWS CLI Comandos.

confirm-sign-up

O código de exemplo a seguir mostra como usar `confirm-sign-up`.

AWS CLI

Como confirmar a inscrição

Este exemplo confirma a inscrição para o nome de usuário `diego@example.com`.

Comando:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --  
username=diego@example.com --confirmation-code CONF_CODE
```

- Para API obter detalhes, consulte [ConfirmSignUp](#) na Referência de AWS CLI Comandos.

create-group

O código de exemplo a seguir mostra como usar `create-group`.

AWS CLI

Para criar um grupo

Este exemplo cria um grupo com uma descrição.

Comando:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaa --group-name MyNewGroup --description "New group."
```

Saída:

```
{
  "Group": {
    "GroupName": "MyNewGroup",
    "UserPoolId": "us-west-2_aaaaaaaa",
    "Description": "New group.",
    "LastModifiedDate": 1548270073.795,
    "CreationDate": 1548270073.795
  }
}
```

Para criar um grupo com uma função e precedência

Este exemplo cria um grupo com uma descrição. Também inclui uma função e precedência.

Comando:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaa --group-name MyNewGroupWithRole --description "New group with a role." --role-arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

Saída:

```
{
  "Group": {
    "GroupName": "MyNewGroupWithRole",
    "UserPoolId": "us-west-2_aaaaaaaa",
    "Description": "New group with a role.",
    "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
    "Precedence": 2,
    "LastModifiedDate": 1548270211.761,
    "CreationDate": 1548270211.761
  }
}
```

- Para API obter detalhes, consulte [CreateGroup](#) na Referência de AWS CLI Comandos.

create-user-import-job

O código de exemplo a seguir mostra como usar `create-user-import-job`.

AWS CLI

Para criar um trabalho de importação de usuários

Este exemplo cria um trabalho de importação de usuário chamado `MyImportJob`.

Para obter mais informações sobre a importação de usuários, consulte [Importação de usuários para grupos de usuários a partir de um CSV arquivo](#).

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
```

Saída:

```
{  
  "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  
  }  
}
```

Faça o upload do `arquivo.csv` com `curl` usando o arquivo pré-assinado: URL

Comando:

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"  
"PRE_SIGNED_URL"
```

- Para API obter detalhes, consulte [CreateUserImportJob](#) na Referência de AWS CLI Comandos.

create-user-pool-client

O código de exemplo a seguir mostra como usar `create-user-pool-client`.

AWS CLI

Para criar um cliente de grupo de usuários

Este exemplo cria um novo cliente de grupo de usuários com dois fluxos de autorização explícitos: `USER_PASSWORD_AUTH` e `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"
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [CreateUserPoolClient](#) na Referência de AWS CLI Comandos.

create-user-pool-domain

O código de exemplo a seguir mostra como usar `create-user-pool-domain`.

AWS CLI

Para criar um domínio de grupo de usuários

Este exemplo cria um novo domínio de grupo de usuários. com dois fluxos de autorização explícitos: USER __ AUTH e PASSWORD ADMIN _NO_ __. SRP AUTH

Comando:

```
aws cognito-idp create-user-pool-domain --user-pool-id us-west-2_aaaaaaaaa --  
domain my-new-domain
```

- Para API obter detalhes, consulte [CreateUserPoolDomain](#) na Referência de AWS CLI Comandos.

create-user-pool

O código de exemplo a seguir mostra como usar create-user-pool.

AWS CLI

Como criar um grupo de usuários minimamente configurado

Este exemplo cria um grupo de usuários chamado MyUserPool usando valores padrão. Não há atributos nem clientes da aplicação obrigatórios. MFA e a segurança avançada está desativada.

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Saída:

```
{  
  "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_aaaaaaaaaa",
  "LambdaConfig": {}
}
}

```

Como criar um grupo de usuários com dois atributos obrigatórios

Este exemplo cria um grupo de usuários MyUserPool. O grupo é configurado para aceitar o e-mail como o atributo de nome de usuário. Ele também define o endereço de origem do e-mail como um endereço validado usando o Amazon Simple Email Service.

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"

```

Saída:

```

{
  "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_aaaaaaaa",  
"LambdaConfig": {}  
}  
}
```

- Para API obter detalhes, consulte [CreateUserPool](#) na Referência de AWS CLI Comandos.

delete-group

O código de exemplo a seguir mostra como usar `delete-group`.

AWS CLI

Para excluir um grupo

Este exemplo exclui um grupo.

Comando:

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaa --group-  
name MyGroupName
```

- Para API obter detalhes, consulte [DeleteGroup](#) na Referência de AWS CLI Comandos.

delete-identity-provider

O código de exemplo a seguir mostra como usar `delete-identity-provider`.

AWS CLI

Para excluir um provedor de identidade

Este exemplo exclui um provedor de identidade.

Comando:

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaa --  
provider-name Facebook
```

- Para API obter detalhes, consulte [DeleteIdentityProvider](#) na Referência de AWS CLI Comandos.

delete-resource-server

O código de exemplo a seguir mostra como usar `delete-resource-server`.

AWS CLI

Para excluir um servidor de recursos

Este exemplo exclui um servidor de recursos chamado `weather.example.com`.

Comando:

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --  
identifier weather.example.com
```

- Para API obter detalhes, consulte [DeleteResourceServer](#) na Referência de AWS CLI Comandos.

delete-user-attributes

O código de exemplo a seguir mostra como usar `delete-user-attributes`.

AWS CLI

Para excluir atributos do usuário

Este exemplo exclui o atributo do usuário "FAVORITE_ANIMAL".

Comando:

```
aws cognito-idp delete-user-attributes --access-token ACCESS_TOKEN --user-attribute-  
names "FAVORITE_ANIMAL"
```

- Para API obter detalhes, consulte [DeleteUserAttributes](#) na Referência de AWS CLI Comandos.

delete-user-pool-client

O código de exemplo a seguir mostra como usar `delete-user-pool-client`.

AWS CLI

Para excluir um cliente de grupo de usuários

Este exemplo exclui um cliente de grupo de usuários.

Comando:

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kpbsnet7mp1d0
```

- Para API obter detalhes, consulte [DeleteUserPoolClient](#) na Referência de AWS CLI Comandos.

delete-user-pool-domain

O código de exemplo a seguir mostra como usar delete-user-pool-domain.

AWS CLI

Para excluir um domínio do grupo de usuários

O delete-user-pool-domain exemplo a seguir exclui um domínio do grupo de usuários chamado my-domain

```
aws cognito-idp delete-user-pool-domain \
  --user-pool-id us-west-2_aaaaaaaaa \
  --domain my-domain
```

- Para API obter detalhes, consulte [DeleteUserPoolDomain](#) na Referência de AWS CLI Comandos.

delete-user-pool

O código de exemplo a seguir mostra como usar delete-user-pool.

AWS CLI

Para excluir um grupo de usuários

Este exemplo exclui um grupo de usuários usando o ID do grupo de usuários, us-west-2_aaaaaaaaa.

Comando:

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- Para API obter detalhes, consulte [DeleteUserPool](#) na Referência de AWS CLI Comandos.

delete-user

O código de exemplo a seguir mostra como usar delete-user.

AWS CLI

Como excluir um usuário

Este exemplo exclui um usuário.

Comando:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

describe-identity-provider

O código de exemplo a seguir mostra como usar describe-identity-provider.

AWS CLI

Para descrever um provedor de identidade

Este exemplo descreve um provedor de identidade chamado Facebook.

Comando:

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --  
provider-name Facebook
```

Saída:

```
{  
  "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
}
}

```

- Para API obter detalhes, consulte [DescribeIdentityProvider](#) na Referência de AWS CLI Comandos.

describe-resource-server

O código de exemplo a seguir mostra como usar `describe-resource-server`.

AWS CLI

Para descrever um servidor de recursos

Este exemplo descreve o servidor de recursos `weather.example.com`.

Comando:

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

Saída:

```

{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Weather",
    "Scopes": [

```

```
    {
      "ScopeName": "weather.update",
      "ScopeDescription": "Update weather forecast"
    },
    {
      "ScopeName": "weather.read",
      "ScopeDescription": "Read weather forecasts"
    },
    {
      "ScopeName": "weather.delete",
      "ScopeDescription": "Delete a weather forecast"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeResourceServer](#) na Referência de AWS CLI Comandos.

describe-risk-configuration

O código de exemplo a seguir mostra como usar `describe-risk-configuration`.

AWS CLI

Para descrever uma configuração de risco

Este exemplo descreve a configuração de risco associada ao pool `us-west-2_aaaaaaaaaa`.

Comando:

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaaa
```

Saída:

```
{
  "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": {

```


AWS CLI

Para descrever um trabalho de importação de usuários

Este exemplo descreve um trabalho de entrada do usuário.

Para obter mais informações sobre a importação de usuários, consulte [Importação de usuários para grupos de usuários a partir de um CSV arquivo](#).

Comando:

```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

Saída:

```
{
  "UserImportJob": {
    "JobName": "import-Test1",
    "JobId": "import-TZqNQvDRnW",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED URL",
    "CreationDate": 1548271708.512,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/
CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

- Para API obter detalhes, consulte [DescribeUserImportJob](#) na Referência de AWS CLI Comandos.

describe-user-pool-client

O código de exemplo a seguir mostra como usar `describe-user-pool-client`.

AWS CLI

Para descrever um cliente de grupo de usuários

Este exemplo descreve um cliente de grupo de usuários.

Comando:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --  
client-id 38fjsnc484p94kpbsnet7mpld0
```

Saída:

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "ClientName": "MyApp",  
    "ClientId": "38fjsnc484p94kpbsnet7mpld0",  
    "ClientSecret": "CLIENT_SECRET",  
    "LastModifiedDate": 1548108676.163,  
    "CreationDate": 1548108676.163,  
    "RefreshTokenValidity": 30,  
    "ReadAttributes": [  
      "address",  
      "birthdate",  
      "custom:CustomAttr1",  
      "custom:CustomAttr2",  
      "email",  
      "email_verified",  
      "family_name",  
      "gender",  
      "given_name",  
      "locale",  
      "middle_name",  
      "name",  
      "nickname",  
      "phone_number",  
      "phone_number_verified",  
      "picture",  
      "preferred_username",  
      "profile",  
      "updated_at",  
      "website",  
      "zoneinfo"  
    ],  
    "WriteAttributes": [  
      "address",
```

```
    "birthdate",
    "custom:CustomAttr1",
    "custom:CustomAttr2",
    "email",
    "family_name",
    "gender",
    "given_name",
    "locale",
    "middle_name",
    "name",
    "nickname",
    "phone_number",
    "picture",
    "preferred_username",
    "profile",
    "updated_at",
    "website",
    "zoneinfo"
  ],
  "ExplicitAuthFlows": [
    "ADMIN_NO_SRP_AUTH",
    "USER_PASSWORD_AUTH"
  ],
  "AllowedOAuthFlowsUserPoolClient": false
}
}
```

- Para API obter detalhes, consulte [DescribeUserPoolClient](#) na Referência de AWS CLI Comandos.

describe-user-pool-domain

O código de exemplo a seguir mostra como usar `describe-user-pool-domain`.

AWS CLI

Para descrever um cliente de grupo de usuários

Este exemplo descreve um domínio de grupo de usuários chamado `my-domain`.

Comando:

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

Saída:

```
{
  "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": {}
  }
}
```

- Para API obter detalhes, consulte [DescribeUserPoolDomain](#) na Referência de AWS CLI Comandos.

describe-user-pool

O código de exemplo a seguir mostra como usar `describe-user-pool`.

AWS CLI

Para descrever um grupo de usuários

Este exemplo descreve um grupo de usuários com o ID de grupo de usuários `us-west-2_aaaaaaaaa`.

Comando:

```
aws cognito-idp describe-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

Saída:

```
{
  "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": 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_aaaaaaaaa",
"LambdaConfig": {}
}
}

```

- Para API obter detalhes, consulte [DescribeUserPool](#) na Referência de AWS CLI Comandos.

forget-device

O código de exemplo a seguir mostra como usar forget-device.

AWS CLI

Para esquecer um dispositivo

Este exemplo esquece o dispositivo como um dispositivo.

Comando:

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- Para API obter detalhes, consulte [ForgetDevice](#) na Referência de AWS CLI Comandos.

forgot-password

O código de exemplo a seguir mostra como usar forgot-password.

AWS CLI

Para forçar uma alteração de senha

O forgot-password exemplo a seguir envia uma mensagem para jane@example.com para alterar a senha.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpbsnet7mpld0 --  
username jane@example.com
```

Saída:

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

- Para API obter detalhes, consulte [ForgotPassword](#) na Referência de AWS CLI Comandos.

get-csv-header

O código de exemplo a seguir mostra como usar get-csv-header.

AWS CLI

Para criar um cabeçalho csv

Este exemplo cria um cabeçalho csv.

Para obter mais informações sobre a importação de usuários, consulte [Importação de usuários para grupos de usuários a partir de um CSV arquivo](#).

Comando:

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaaa
```

Saída:

```
{
  "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"
  ]
}
```

... Importação de usuários para grupos de usuários a partir de um CSV arquivo: <https://docs.aws.amazon.com/cognito/cognito-user-pools-using-latest/developerguide/-import-tool.html>

- Para API obter detalhes, consulte [GetCsvHeader](#) na Referência de AWS CLI Comandos.

get-group

O código de exemplo a seguir mostra como usar `get-group`.

AWS CLI

Para obter informações sobre um grupo

Este exemplo obtém informações sobre um grupo chamado MyGroup.

Comando:

```
aws cognito-idp get-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

Saída:

```
{
  "Group": {
    "GroupName": "MyGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "A sample group.",
    "LastModifiedDate": 1548270073.795,
    "CreationDate": 1548270073.795
  }
}
```

- Para API obter detalhes, consulte [GetGroup](#) na Referência de AWS CLI Comandos.

get-signing-certificate

O código de exemplo a seguir mostra como usar `get-signing-certificate`.

AWS CLI

Para obter um certificado de assinatura

Este exemplo obtém um certificado de assinatura para um grupo de usuários.

Comando:

```
aws cognito-idp get-signing-certificate --user-pool-id us-west-2_aaaaaaaaa
```

Saída:

```
{
  "Certificate": "CERTIFICATE_DATA"
}
```

- Para API obter detalhes, consulte [GetSigningCertificate](#) na Referência de AWS CLI Comandos.

get-ui-customization

O código de exemplo a seguir mostra como usar `get-ui-customization`.

AWS CLI

Para obter informações de personalização da interface do usuário

Este exemplo obtém informações de personalização da interface do usuário para um grupo de usuários.

Comando:

```
aws cognito-idp get-ui-customization --user-pool-id us-west-2_aaaaaaaaa
```

Saída:

```
{
  "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\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
\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
\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.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"
  }
}

```

- Para API obter detalhes, consulte [GetUiCustomization](#) na Referência de AWS CLI Comandos.

list-user-import-jobs

O código de exemplo a seguir mostra como usar `list-user-import-jobs`.

AWS CLI

Para listar trabalhos de importação de usuários

Este exemplo lista trabalhos de importação de usuários.

Para obter mais informações sobre a importação de usuários, consulte [Importação de usuários para grupos de usuários a partir de um CSV arquivo](#).

Comando:

```
aws cognito-idp list-user-import-jobs --user-pool-id us-west-2_aaaaaaaaaa --max-
results 20
```

Saída:

```
{
  "UserImportJobs": [
    {
      "JobName": "Test2",
      "JobId": "import-d00nwGA3mV",

```

```

    "UserPoolId": "us-west-2_aaaaaaaaa",
    "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."
  }
]
}

```

- Para API obter detalhes, consulte [ListUserImportJobs](#) na Referência de AWS CLI Comandos.

list-user-pools

O código de exemplo a seguir mostra como usar `list-user-pools`.

AWS CLI

Como listar grupos de usuários

Este exemplo lista até vinte grupos de usuários.

Comando:

```
aws cognito-idp list-user-pools --max-results 20
```

Saída:

```
{
  "UserPools": [
    {
      "CreationDate": 1547763720.822,
      "LastModifiedDate": 1547763720.822,
      "LambdaConfig": {},
      "Id": "us-west-2_aaaaaaaaaa",
      "Name": "MyUserPool"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListUserPools](#) na Referência de AWS CLI Comandos.

list-users-in-group

O código de exemplo a seguir mostra como usar `list-users-in-group`.

AWS CLI

Para listar usuários em um grupo

Este exemplo lista os usuários em um grupo MyGroup.

Comando:

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

Saída:

```
{
  "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"
  }
]
}
```

- Para API obter detalhes, consulte [ListUsersInGroup](#) na Referência de AWS CLI Comandos.

list-users

O código de exemplo a seguir mostra como usar `list-users`.

AWS CLI

Como listar usuários

Este exemplo lista até vinte usuários.

Comando:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

Saída:

```
{
  "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"
      }
    ]
  }
]
}

```

- Para API obter detalhes, consulte [ListUsers](#) na Referência de AWS CLI Comandos.

resend-confirmation-code

O código de exemplo a seguir mostra como usar `resend-confirmation-code`.

AWS CLI

Como reenviar um código de confirmação

O exemplo `resend-confirmation-code` a seguir envia um código de confirmação ao usuário `jane`.

```

aws cognito-idp resend-confirmation-code \
  --client-id 12a3b456c7de890f11g123hijk \
  --username jane

```

Saída:

```

{
  "CodeDeliveryDetails": {
    "Destination": "j***@e***.com",
    "DeliveryMedium": "EMAIL",
    "AttributeName": "email"
  }
}

```

Para obter mais informações, consulte [Como cadastrar e confirmar contas de usuários](#) no Guia do desenvolvedor do Amazon Cognito.

- Para API obter detalhes, consulte [ResendConfirmationCode](#) na Referência de AWS CLI Comandos.

respond-to-auth-challenge

O código de exemplo a seguir mostra como usar `respond-to-auth-challenge`.

AWS CLI

Como responder a um desafio de autorização

Este exemplo responde a um desafio de autorização iniciado com `initiate-auth`. É uma resposta ao `REQUIRED` desafio `NEW PASSWORD` __. Ele define uma senha para o usuário `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"
```

Saída:

```
{
  "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"
    }
  }
}
```

- Para API obter detalhes, consulte [RespondToAuthChallenge](#) na Referência de AWS CLI Comandos.

set-risk-configuration

O código de exemplo a seguir mostra como usar `set-risk-configuration`.

AWS CLI

Para definir a configuração de risco

Este exemplo define a configuração de risco para um grupo de usuários. Ele define a ação do evento de inscrição como `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}
```

Saída:

```
{  
  "RiskConfiguration": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "CompromisedCredentialsRiskConfiguration": {  
      "EventFilter": [  
        "SIGN_UP"  
      ],  
      "Actions": {  
        "EventAction": "NO_ACTION"  
      }  
    }  
  }  
}
```

- Para API obter detalhes, consulte [SetRiskConfiguration](#) na Referência de AWS CLI Comandos.

set-ui-customization

O código de exemplo a seguir mostra como usar `set-ui-customization`.

AWS CLI

Para definir a personalização da interface do usuário

Este exemplo personaliza a CSS configuração de um grupo de usuários.

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.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"
```

Saída:

```
{
  "UICustomization": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientId": "ALL",
    "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": "20190129172214"
  }
}

```

- Para API obter detalhes, consulte [SetUiCustomization](#) na Referência de AWS CLI Comandos.

set-user-mfa-preference

O código de exemplo a seguir mostra como usar `set-user-mfa-preference`.

AWS CLI

Para definir as MFA configurações do usuário

O `set-user-mfa-preference` exemplo a seguir modifica as opções MFA de entrega. Ele muda o meio MFA de entrega para 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

```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar MFA a um grupo de usuários no Guia](#) do desenvolvedor do Amazon Cognito.

- Para API obter detalhes, consulte [SetUserMfaPreference](#) na Referência de AWS CLI Comandos.

set-user-settings

O código de exemplo a seguir mostra como usar `set-user-settings`.

AWS CLI

Para definir as configurações do usuário

Este exemplo define a preferência MFA de entrega como EMAIL.

Comando:

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-  
options DeliveryMedium=EMAIL
```

- Para API obter detalhes, consulte [SetUserSettings](#) na Referência de AWS CLI Comandos.

sign-up

O código de exemplo a seguir mostra como usar `sign-up`.

AWS CLI

Como inscrever um usuário

Este exemplo inscreve `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"
```

Saída:

```
{  
  "UserConfirmed": false,
```

```
"UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- Para API obter detalhes, consulte [SignUp](#) na Referência de AWS CLI Comandos.

start-user-import-job

O código de exemplo a seguir mostra como usar `start-user-import-job`.

AWS CLI

Para iniciar um trabalho de importação de usuários

Este exemplo inicia um trabalho de entrada do usuário.

Para obter mais informações sobre a importação de usuários, consulte [Importação de usuários para grupos de usuários a partir de um CSV](#) arquivo.

Comando:

```
aws cognito-idp start-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

Saída:

```
{
  "UserImportJob": {
    "JobName": "import-Test10",
    "JobId": "import-lmpxS0uIzH",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED_URL",
    "CreationDate": 1548278378.928,
    "StartDate": 1548278397.334,
    "Status": "Pending",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

- Para API obter detalhes, consulte [StartUserImportJob](#) na Referência de AWS CLI Comandos.

stop-user-import-job

O código de exemplo a seguir mostra como usar `stop-user-import-job`.

AWS CLI

Para interromper um trabalho de importação de usuários

Este exemplo interrompe um trabalho de entrada do usuário.

Para obter mais informações sobre a importação de usuários, consulte [Importação de usuários para grupos de usuários a partir de um CSV arquivo](#).

Comando:

```
aws cognito-idp stop-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-id import-TZqNQvDRnW
```

Saída:

```
{
  "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."
  }
}
```

- Para API obter detalhes, consulte [StopUserImportJob](#) na Referência de AWS CLI Comandos.

update-auth-event-feedback

O código de exemplo a seguir mostra como usar `update-auth-event-feedback`.

AWS CLI

Para atualizar o feedback do evento de autenticação

Este exemplo atualiza o feedback do evento de autorização. Isso marca o evento como “Válido”.

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"
```

- Para API obter detalhes, consulte [UpdateAuthEventFeedback](#) na Referência de AWS CLI Comandos.

update-device-status

O código de exemplo a seguir mostra como usar `update-device-status`.

AWS CLI

Para atualizar o status do dispositivo

Este exemplo atualiza o status de um dispositivo para “not_remembered”.

Comando:

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-key DEVICE_KEY --device-remembered-status "not_remembered"
```

- Para API obter detalhes, consulte [UpdateDeviceStatus](#) na Referência de AWS CLI Comandos.

update-group

O código de exemplo a seguir mostra como usar `update-group`.

AWS CLI

Para atualizar um grupo

Este exemplo atualiza a descrição e a precedência de MyGroup.

Comando:

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
--description "New description" --precedence 2
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [UpdateGroup](#) na Referência de AWS CLI Comandos.

update-resource-server

O código de exemplo a seguir mostra como usar update-resource-server.

AWS CLI

Para atualizar um servidor de recursos

Este exemplo atualiza o servidor de recursos Weather. Ele adiciona um novo escopo.

Comando:

```
aws cognito-idp update-resource-server --user-pool-id us-west-2_aaaaaaaaa
--identifier weather.example.com --name Weather --scopes
ScopeName=NewScope,ScopeDescription="New scope description"
```

Saída:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Happy",
    "Scopes": [
      {
        "ScopeName": "NewScope",
        "ScopeDescription": "New scope description"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [UpdateResourceServer](#) na Referência de AWS CLI Comandos.

update-user-attributes

O código de exemplo a seguir mostra como usar `update-user-attributes`.

AWS CLI

Para atualizar os atributos do usuário

Este exemplo atualiza o atributo de usuário “apelido”.

Comando:

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
Name="nickname",Value="Dan"
```

- Para API obter detalhes, consulte [UpdateUserAttributes](#) na Referência de AWS CLI Comandos.

update-user-pool-client

O código de exemplo a seguir mostra como usar `update-user-pool-client`.

AWS CLI

Para atualizar um cliente de grupo de usuários

Este exemplo atualiza o nome de um cliente de grupo de usuários. Ele também adiciona um atributo gravável “apelido”.

Comando:

```
aws cognito-idp update-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --  
client-id 3n4b5urk1ft4fl3mg5e62d9ado --client-name "NewClientName" --write-  
attributes "nickname"
```

Saída:

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "ClientName": "NewClientName",  
    "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",  
    "LastModifiedDate": 1548802761.334,  
    "CreationDate": 1548178931.258,  
    "RefreshTokenValidity": 30,  
    "WriteAttributes": [  
      "nickname"  
    ],  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```

- Para API obter detalhes, consulte [UpdateUserPoolClient](#) na Referência de AWS CLI Comandos.

update-user-pool

O código de exemplo a seguir mostra como usar `update-user-pool`.

AWS CLI

Para atualizar um grupo de usuários

Este exemplo adiciona tags a um grupo de usuários.

Comando:

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_aaaaaaaaa --user-pool-  
tags Team=Blue,Area=West
```

- Para API obter detalhes, consulte [UpdateUserPool](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Comprehend usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Comprehend.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-detect-dominant-language

O código de exemplo a seguir mostra como usar `batch-detect-dominant-language`.

AWS CLI

Detectar o idioma dominante de vários textos de entrada

O `batch-detect-dominant-language` exemplo a seguir analisa vários textos de entrada e retorna o idioma dominante de cada um. A pontuação de confiança dos modelos pré-treinados também é gerada para cada previsão.

```
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."
```

Saída:

```
{  
  "ResultList": [  

```

```

    {
      "Index": 0,
      "Languages": [
        {
          "LanguageCode": "en",
          "Score": 0.9986501932144165
        }
      ]
    }
  ],
  "ErrorList": []
}

```

Para obter mais informações, consulte [Idioma dominante](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [BatchDetectDominantLanguage](#) na Referência de AWS CLI Comandos.

batch-detect-entities

O código de exemplo a seguir mostra como usar batch-detect-entities.

AWS CLI

Para detectar entidades de vários textos de entrada

O batch-detect-entities exemplo a seguir analisa vários textos de entrada e retorna as entidades nomeadas de cada um. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```

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."

```

Saída:

```

{
  "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": []
}
```

Para obter mais informações, consulte [Entidades](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [BatchDetectEntities](#) na Referência de AWS CLI Comandos.

batch-detect-key-phrases

O código de exemplo a seguir mostra como usar `batch-detect-key-phrases`.

AWS CLI

Para detectar frases-chave de várias entradas de texto

O `batch-detect-key-phrases` exemplo a seguir analisa vários textos de entrada e retorna as frases nominais principais de cada um. A pontuação de confiança do modelo pré-treinado para cada previsão também é gerada.

```
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."
```

Saída:

```
{  
  "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": 61  
        }  
      ]  
    }  
  ]  
}
```

```
        "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": []
```



```
}
```

Para obter mais informações, consulte [Frases-chave](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [BatchDetectKeyPhrases](#) na Referência de AWS CLI Comandos.

batch-detect-sentiment

O código de exemplo a seguir mostra como usar batch-detect-sentiment.

AWS CLI

Para detectar o sentimento predominante de vários textos de entrada

O batch-detect-sentiment exemplo a seguir analisa vários textos de entrada e retorna o sentimento predominante (POSITIVE,, ou NEUTRAL MIXEDNEGATIVE, de cada um).

```
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
```

Saída:

```
{  
  "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": []
}

```

Para obter mais informações, consulte [Sentiment](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [BatchDetectSentiment](#) na Referência de AWS CLI Comandos.

batch-detect-syntax

O código de exemplo a seguir mostra como usar batch-detect-syntax.

AWS CLI

Para inspecionar a sintaxe e partes da fala de palavras em vários textos de entrada

O batch-detect-syntax exemplo a seguir analisa a sintaxe de vários textos de entrada e retorna as diferentes partes da fala. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```

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

```

Saída:

```
{
  "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": []
}

```

Para obter mais informações, consulte [Análise de sintaxe](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [BatchDetectSyntax](#) na Referência de AWS CLI Comandos.

batch-detect-targeted-sentiment

O código de exemplo a seguir mostra como usar batch-detect-targeted-sentiment.

AWS CLI

Para detectar o sentimento e cada entidade nomeada para vários textos de entrada

O batch-detect-targeted-sentiment exemplo a seguir analisa vários textos de entrada e retorna as entidades nomeadas junto com o sentimento predominante associado a cada entidade. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```

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."

```

Saída:

```

{
  "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": 9.99999974752427e-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": 9.0000000318337698e-06,
                "Negative": 1.9999999949504854e-06,
                "Neutral": 0.9999859929084778,
                "Mixed": 3.999999989900971e-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": []
}

```

Para obter mais informações, consulte [Sentimento direcionado](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [BatchDetectTargetedSentiment](#) na Referência de AWS CLI Comandos.

classify-document

O código de exemplo a seguir mostra como usar `classify-document`.

AWS CLI

Para classificar o documento com um endpoint específico do modelo

O `classify-document` exemplo a seguir classifica um documento com um endpoint de um modelo personalizado. O modelo neste exemplo foi treinado em um conjunto de dados contendo mensagens SMS rotuladas como spam ou não spam, ou “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"
```

Saída:

```
{  
  "Classes": [  
    {  
      "Name": "spam",  
      "Score": 0.9998599290847778  
    },  
    {  
      "Name": "ham",  
      "Score": 0.00014001205272506922  
    }  
  ]  
}
```

Para obter mais informações, consulte [Classificação personalizada](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ClassifyDocument](#) na Referência de AWS CLI Comandos.

contains-pii-entities

O código de exemplo a seguir mostra como usar `contains-pii-entities`.

AWS CLI

Analisar o texto de entrada quanto à presença de PII informações

O `contains-pii-entities` exemplo a seguir analisa o texto de entrada para detectar a presença de informações de identificação pessoal (PII) e retorna os rótulos dos tipos de PII entidade identificados, como nome, endereço, número da conta bancária ou número de telefone.

```
aws comprehend contains-pii-entities \  
  --text "123 Main St, New York, NY 10001"
```

```
--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."
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Informações de identificação pessoal \(PII\)](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ContainsPiiEntities](#) na Referência de AWS CLI Comandos.

create-dataset

O código de exemplo a seguir mostra como usar create-dataset.

AWS CLI

Para criar um conjunto de dados do flywheel

O create-dataset exemplo a seguir cria um conjunto de dados para um volante. Esse conjunto de dados será usado como dados adicionais de treinamento, conforme especificado pela --dataset-type tag.

```
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
```

Conteúdo de file://inputConfig.json:

```
{  
  "DataFormat": "COMPREHEND_CSV",  
  "DocumentClassifierInputDataConfig": {  
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/training-data.csv"  
  }  
}
```

Saída:

```
{  
  "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset"  
}
```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [CreateDataset](#) na Referência de AWS CLI Comandos.

create-document-classifier

O código de exemplo a seguir mostra como usar `create-document-classifier`.

AWS CLI

Para criar um classificador de documentos para categorizar documentos

O exemplo de `create-document-classifier` a seguir inicia o processo de treinamento para um modelo de classificador de documentos. O arquivo de dados de treinamento, `training.csv`, está localizado na tag `--input-data-config`. `training.csv` é um documento de duas colunas em que os rótulos ou classificações são fornecidos na primeira coluna e os documentos são fornecidos na segunda coluna.

```
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
```

Saída:

```
{  
  "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier"  
}
```

Para obter mais informações, consulte [Classificação personalizada](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [CreateDocumentClassifier](#) na Referência de AWS CLI Comandos.

create-endpoint

O código de exemplo a seguir mostra como usar `create-endpoint`.

AWS CLI

Para criar um endpoint para um modelo personalizado

O `create-endpoint` exemplo a seguir cria um endpoint para inferência síncrona para um modelo personalizado previamente treinado.

```
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
```

Saída:

```
{  
  "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint-1"  
}
```

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [CreateEndpoint](#) na Referência de AWS CLI Comandos.

create-entity-recognizer

O código de exemplo a seguir mostra como usar `create-entity-recognizer`.

AWS CLI

Para criar um reconhecedor de entidades personalizado

O `create-entity-recognizer` exemplo a seguir inicia o processo de treinamento para um modelo personalizado de reconhecimento de entidades. Este exemplo usa um CSV arquivo contendo documentos de `raw_text.csv` treinamento e uma lista de CSV entidades `entity_list.csv` para treinar o modelo. `entity-list.csv` contém as seguintes colunas: texto 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
```

Saída:

```
{
  "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-
entity-recognizer/entityrecognizer1"
}
```

Para obter mais informações, consulte [Reconhecimento personalizado de entidades](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [CreateEntityRecognizer](#) na Referência de AWS CLI Comandos.

create-flywheel

O código de exemplo a seguir mostra como usar `create-flywheel`.

AWS CLI

Para criar um volante

O `create-flywheel` exemplo a seguir cria um volante para orquestrar o treinamento contínuo de um modelo de classificação de documentos ou de reconhecimento de entidades. O volante neste exemplo foi criado para gerenciar um modelo treinado existente especificado pela `--active-model-arn` tag. Quando o volante é criado, um lago de dados é criado na `--input-data-lake` tag.

```
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"
```

Saída:

```
{
```

```
"FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel"
}
```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [CreateFlywheel](#) na Referência de AWS CLI Comandos.

delete-document-classifier

O código de exemplo a seguir mostra como usar `delete-document-classifier`.

AWS CLI

Para excluir um classificador de documentos personalizado

O exemplo de `delete-document-classifier` a seguir exclui um modelo de classificador de documentos personalizado.

```
aws comprehend delete-document-classifier \
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DeleteDocumentClassifier](#) na Referência de AWS CLI Comandos.

delete-endpoint

O código de exemplo a seguir mostra como usar `delete-endpoint`.

AWS CLI

Para excluir um endpoint de um modelo personalizado

O `delete-endpoint` exemplo a seguir exclui um endpoint específico do modelo. Todos os endpoints devem ser excluídos para que o modelo seja excluído.

```
aws comprehend delete-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint-1
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DeleteEndpoint](#) na Referência de AWS CLI Comandos.

delete-entity-recognizer

O código de exemplo a seguir mostra como usar delete-entity-recognizer.

AWS CLI

Para excluir um modelo de reconhecimento de entidade personalizado

O delete-entity-recognizer exemplo a seguir exclui um modelo personalizado de reconhecimento de entidades.

```
aws comprehend delete-entity-recognizer \  
  --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-  
recognizer/example-entity-recognizer-1
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DeleteEntityRecognizer](#) na Referência de AWS CLI Comandos.

delete-flywheel

O código de exemplo a seguir mostra como usar delete-flywheel.

AWS CLI

Para excluir um volante

O `delete-flywheel` exemplo a seguir exclui um volante. O data lake ou o modelo associado ao volante não é excluído.

```
aws comprehend delete-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel-1
```

Este comando não produz saída.

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [DeleteFlywheel](#) na Referência de AWS CLI Comandos.

delete-resource-policy

O código de exemplo a seguir mostra como usar `delete-resource-policy`.

AWS CLI

Para excluir uma política baseada em recursos

O `delete-resource-policy` exemplo a seguir exclui uma política baseada em recursos de um recurso do Amazon Comprehend.

```
aws comprehend delete-resource-policy \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier-1/version/1
```

Este comando não produz saída.

Para obter mais informações, consulte [Cópia de modelos personalizados entre AWS contas](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DeleteResourcePolicy](#) na Referência de AWS CLI Comandos.

describe-dataset

O código de exemplo a seguir mostra como usar `describe-dataset`.

AWS CLI

Para descrever um conjunto de dados do volante

O `describe-dataset` exemplo a seguir obtém as propriedades de um conjunto de dados do flywheel.

```
aws comprehend describe-dataset \  
  --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [DescribeDataset](#) na Referência de AWS CLI Comandos.

describe-document-classification-job

O código de exemplo a seguir mostra como usar `describe-document-classification-job`.

AWS CLI

Para descrever um trabalho de classificação de documentos

O exemplo de `describe-document-classification-job` a seguir obtém as propriedades de um trabalho assíncrono de classificação de documentos.

```
aws comprehend describe-document-classification-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Classificação personalizada](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeDocumentClassificationJob](#) na Referência de AWS CLI Comandos.

describe-document-classifier

O código de exemplo a seguir mostra como usar `describe-document-classifier`.

AWS CLI

Para descrever um classificador de documentos

O exemplo de `describe-document-classifier` a seguir obtém as propriedades de um modelo de classificador de documentos personalizado.

```
aws comprehend describe-document-classifier \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-1
```

Saída:

```
{  
  "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"  
  }  
}
```



```
}
```

Para obter mais informações, consulte [Criação e gerenciamento de modelos personalizados](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeDocumentClassifier](#) na Referência de AWS CLI Comandos.

describe-dominant-language-detection-job

O código de exemplo a seguir mostra como usar `describe-dominant-language-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de linguagem dominante.

O `describe-dominant-language-detection-job` exemplo a seguir obtém as propriedades de um trabalho assíncrono de detecção de linguagem dominante.

```
aws comprehend describe-dominant-language-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeDominantLanguageDetectionJob](#) na Referência de AWS CLI Comandos.

describe-endpoint

O código de exemplo a seguir mostra como usar `describe-endpoint`.

AWS CLI

Para descrever um endpoint específico

O `describe-endpoint` exemplo a seguir obtém as propriedades de um endpoint específico do modelo.

```

aws comprehend describe-endpoint \
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint

```

Saída:

```

{
  "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"
  }
}

```

```
}  
}
```

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeEndpoint](#) na Referência de AWS CLI Comandos.

describe-entities-detection-job

O código de exemplo a seguir mostra como usar `describe-entities-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de entidades

O `describe-entities-detection-job` exemplo a seguir obtém as propriedades de um trabalho de detecção de entidades assíncronas.

```
aws comprehend describe-entities-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeEntitiesDetectionJob](#) na Referência de AWS CLI Comandos.

describe-entity-recognizer

O código de exemplo a seguir mostra como usar `describe-entity-recognizer`.

AWS CLI

Para descrever um reconhecedor de entidades

O `describe-entity-recognizer` exemplo a seguir obtém as propriedades de um modelo personalizado de reconhecimento de entidades.

```
aws comprehend describe-entity-recognizer \
  entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/business-recongizer-1/version/1
```

Saída:

```
{
  "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"
}
}
```

Para obter mais informações, consulte [Reconhecimento personalizado de entidades](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [DescribeEntityRecognizerna](#) Referência de AWS CLI Comandos.

describe-events-detection-job

O código de exemplo a seguir mostra como usar `describe-events-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de eventos.

O `describe-events-detection-job` exemplo a seguir obtém as propriedades de um trabalho de detecção de eventos assíncronos.

```
aws comprehend describe-events-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeEventsDetectionJob](#) na Referência de AWS CLI Comandos.

describe-flywheel-iteration

O código de exemplo a seguir mostra como usar `describe-flywheel-iteration`.

AWS CLI

Para descrever uma iteração do volante

O `describe-flywheel-iteration` exemplo a seguir obtém as propriedades de uma iteração do volante.

```
aws comprehend describe-flywheel-iteration \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel \
  --flywheel-iteration-id 20232222AEXAMPLE
```

Saída:

```
{
  "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/"
}
}

```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [DescribeFlywheelIteration](#) na Referência de AWS CLI Comandos.

describe-flywheel

O código de exemplo a seguir mostra como usar `describe-flywheel`.

AWS CLI

Para descrever um volante

O `describe-flywheel` exemplo a seguir obtém as propriedades de um volante. Neste exemplo, o modelo associado ao volante é um modelo de classificador personalizado treinado para classificar documentos como spam ou não spam, ou como “falsos”.

```

aws comprehend describe-flywheel \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [DescribeFlywheel](#) na Referência de AWS CLI Comandos.

describe-key-phrases-detection-job

O código de exemplo a seguir mostra como usar `describe-key-phrases-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de frases-chave

O `describe-key-phrases-detection-job` exemplo a seguir obtém as propriedades de um trabalho assíncrono de detecção de frases-chave.

```
aws comprehend describe-key-phrases-detection-job \
```

```
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeKeyPhrasesDetectionJob](#) na Referência de AWS CLI Comandos.

describe-pii-entities-detection-job

O código de exemplo a seguir mostra como usar `describe-pii-entities-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de PII entidades

O `describe-pii-entities-detection-job` exemplo a seguir obtém as propriedades de um trabalho assíncrono de detecção de entidades pii.

```
aws comprehend describe-pii-entities-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribePiiEntitiesDetectionJob](#) na Referência de AWS CLI Comandos.

describe-resource-policy

O código de exemplo a seguir mostra como usar `describe-resource-policy`.

AWS CLI

Para descrever uma política de recursos anexada a um modelo

O `describe-resource-policy` exemplo a seguir obtém as propriedades de uma política baseada em recursos anexadas a um modelo.

```
aws comprehend describe-resource-policy \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
  example-classifier/version/1
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Cópia de modelos personalizados entre AWS contas](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeResourcePolicy](#) na Referência de AWS CLI Comandos.

describe-sentiment-detection-job

O código de exemplo a seguir mostra como usar `describe-sentiment-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de sentimentos

O `describe-sentiment-detection-job` exemplo a seguir obtém as propriedades de um trabalho assíncrono de detecção de sentimentos.

```
aws comprehend describe-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeSentimentDetectionJob](#) na Referência de AWS CLI Comandos.

describe-targeted-sentiment-detection-job

O código de exemplo a seguir mostra como usar `describe-targeted-sentiment-detection-job`.

AWS CLI

Para descrever um trabalho direcionado de detecção de sentimentos

O `describe-targeted-sentiment-detection-job` exemplo a seguir obtém as propriedades de um trabalho assíncrono de detecção de sentimentos direcionados.

```

aws comprehend describe-targeted-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE

```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeTargetedSentimentDetectionJob](#) na Referência de AWS CLI Comandos.

describe-topics-detection-job

O código de exemplo a seguir mostra como usar `describe-topics-detection-job`.

AWS CLI

Para descrever um trabalho de detecção de tópicos

O exemplo de `describe-topics-detection-job` a seguir obtém as propriedades de um trabalho assíncrono de detecção de tópicos.

```
aws comprehend describe-topics-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DescribeTopicsDetectionJob](#) na Referência de AWS CLI Comandos.

detect-dominant-language

O código de exemplo a seguir mostra como usar detect-dominant-language.

AWS CLI

Para detectar o idioma dominante do texto de entrada

O detect-dominant-language a seguir analisa o texto de entrada e identifica o idioma dominante. A pontuação de confiança do modelo pré-treinado também é gerada.

```
aws comprehend detect-dominant-language \
```

```
--text "It is a beautiful day in Seattle."
```

Saída:

```
{
  "Languages": [
    {
      "LanguageCode": "en",
      "Score": 0.9877256155014038
    }
  ]
}
```

Para obter mais informações, consulte [Idioma dominante](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DetectDominantLanguage](#) na Referência de AWS CLI Comandos.

detect-entities

O código de exemplo a seguir mostra como usar detect-entities.

AWS CLI

Para detectar entidades nomeadas no texto de entrada

O exemplo de detect-entities a seguir analisa o texto de entrada e retorna as entidades nomeadas. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```
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."
```


Saída:

```
{
  "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
  }
}
```

```
]
}
```

Para obter mais informações, consulte [Entidades](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DetectEntities](#) na Referência de AWS CLI Comandos.

detect-key-phrases

O código de exemplo a seguir mostra como usar `detect-key-phrases`.

AWS CLI

Para detectar frases-chave no texto de entrada

O exemplo de `detect-key-phrases` a seguir analisa o texto de entrada e identifica as principais frases nominais. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```
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."
```

Saída:

```
{
  "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
    }
  ]
}

```

Para obter mais informações, consulte [Frases-chave](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DetectKeyPhrases](#) na Referência de AWS CLI Comandos.

detect-pii-entities

O código de exemplo a seguir mostra como usar `detect-pii-entities`.

AWS CLI

Para detectar entidades PII no texto de entrada

O `detect-pii-entities` exemplo a seguir analisa o texto de entrada e identifica entidades que contêm informações de identificação pessoal (). PII A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```

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."

```

Saída:

```

{
  "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
      }
    ]
  }

```

Para obter mais informações, consulte [Informações de identificação pessoal \(PII\)](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DetectPiiEntities](#) na Referência de AWS CLI Comandos.

detect-sentiment

O código de exemplo a seguir mostra como usar detect-sentiment.

AWS CLI

Para detectar o sentimento de um texto de entrada

O exemplo de detect-sentiment a seguir analisa o texto de entrada e retorna uma inferência do sentimento predominante (POSITIVE, NEUTRAL, MIXED ou NEGATIVE).

```

aws comprehend detect-sentiment \
  --language-code en \
  --text "It is a beautiful day in Seattle"

```

Saída:

```

{
  "Sentiment": "POSITIVE",
  "SentimentScore": {
    "Positive": 0.9976957440376282,
    "Negative": 9.653854067437351e-05,
    "Neutral": 0.002169104292988777,
    "Mixed": 3.857641786453314e-05
  }
}

```


Para obter mais informações, consulte [Sentimento](#) no Guia do desenvolvedor do Amazon Comprehend

- Para API obter detalhes, consulte [DetectSentiment](#) na Referência de AWS CLI Comandos.

detect-syntax

O código de exemplo a seguir mostra como usar detect-syntax.

AWS CLI

Para detectar as partes da fala em um texto de entrada

O exemplo de detect-syntax a seguir analisa a sintaxe do texto de entrada e retorna as diferentes partes da fala. A pontuação de confiança do modelo pré-treinado também é gerada para cada previsão.

```
aws comprehend detect-syntax \  
  --language-code en \  
  --text "It is a beautiful day in Seattle."
```

Saída:

```
{  
  "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
        }
    ]
}
```

Para obter mais informações, consulte [Análise de sintaxe](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DetectSyntax](#) na Referência de AWS CLI Comandos.

detect-targeted-sentiment

O código de exemplo a seguir mostra como usar `detect-targeted-sentiment`.

AWS CLI

Para detectar o sentimento direcionado de entidades nomeadas em um texto de entrada

O `detect-targeted-sentiment` exemplo a seguir analisa o texto de entrada e retorna as entidades nomeadas, além do sentimento direcionado associado a cada entidade. A pontuação de confiança dos modelos pré-treinados para cada previsão também é gerada.

```
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"
```

Saída:

```
{
  "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
  ]
}
```

```

    ]
  }
}

```

Para obter mais informações, consulte [Sentimento direcionado](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [DetectTargetedSentiment](#) na Referência de AWS CLI Comandos.

import-model

O código de exemplo a seguir mostra como usar `import-model`.

AWS CLI

Para importar um modelo

O `import-model` exemplo a seguir importa um modelo de uma AWS conta diferente. O modelo classificador de documentos na conta 444455556666 tem uma política baseada em recursos que permite que 111122223333 a conta importe o modelo.

```

aws comprehend import-model \
  --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-
classifier/example-classifier

```

Saída:

```

{
  "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier"
}

```

Para obter mais informações, consulte [Cópia de modelos personalizados entre AWS contas](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ImportModel](#) na Referência de AWS CLI Comandos.

list-datasets

O código de exemplo a seguir mostra como usar `list-datasets`.

AWS CLI

Para listar todos os conjuntos de dados do flywheel

O `list-datasets` exemplo a seguir lista todos os conjuntos de dados associados a um volante.

```
aws comprehend list-datasets \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [ListDatasets](#) na Referência de AWS CLI Comandos.

list-document-classification-jobs

O código de exemplo a seguir mostra como usar `list-document-classification-jobs`.

AWS CLI

Para listar trabalhos de classificação de documento

O exemplo de `list-document-classification-jobs` a seguir lista todos os trabalhos de classificação de documentos.

```
aws comprehend list-document-classification-jobs
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Classificação personalizada](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListDocumentClassificationJobs](#) na Referência de AWS CLI Comandos.

list-document-classifier-summaries

O código de exemplo a seguir mostra como usar `list-document-classifier-summaries`.

AWS CLI

Para listar os resumos de todos os classificadores de documentos criados

O `list-document-classifier-summaries` exemplo a seguir lista todos os resumos de classificadores de documentos criados.

```
aws comprehend list-document-classifier-summaries
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Criação e gerenciamento de modelos personalizados](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListDocumentClassifierSummaries](#) na Referência de AWS CLI Comandos.

list-document-classifiers

O código de exemplo a seguir mostra como usar `list-document-classifiers`.

AWS CLI

Para listar todos os classificadores de documentos

O exemplo de `list-document-classifiers` a seguir lista todos os modelos de classificadores de documentos treinados e em treinamento.

```
aws comprehend list-document-classifiers
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Criação e gerenciamento de modelos personalizados](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListDocumentClassifiers](#) na Referência de AWS CLI Comandos.

list-dominant-language-detection-jobs

O código de exemplo a seguir mostra como usar `list-dominant-language-detection-jobs`.

AWS CLI

Para listar todos os trabalhos de detecção de linguagem dominante

O `list-dominant-language-detection-jobs` exemplo a seguir lista todos os trabalhos de detecção assíncrona de linguagem dominante em andamento e concluídos.

```
aws comprehend list-dominant-language-detection-jobs
```

Saída:

```

{
  "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"
}
]
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListDominantLanguageDetectionJobs](#) na Referência de AWS CLI Comandos.

list-endpoints

O código de exemplo a seguir mostra como usar `list-endpoints`.

AWS CLI

Para listar todos os endpoints

O `list-endpoints` exemplo a seguir lista todos os endpoints ativos específicos do modelo.

```
aws comprehend list-endpoints
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListEndpoints](#) na Referência de AWS CLI Comandos.

list-entities-detection-jobs

O código de exemplo a seguir mostra como usar `list-entities-detection-jobs`.

AWS CLI

Para listar todas as tarefas de detecção de entidades

O `list-entities-detection-jobs` exemplo a seguir lista todos os trabalhos de detecção de entidades assíncronas.

```
aws comprehend list-entities-detection-jobs
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Entidades](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListEntitiesDetectionJobs](#) na Referência de AWS CLI Comandos.

list-entity-recognizer-summaries

O código de exemplo a seguir mostra como usar `list-entity-recognizer-summaries`.

AWS CLI

Para listar os resumos de todos os reconhedores de entidades criados

O `list-entity-recognizer-summaries` exemplo a seguir lista todos os resumos do reconhedor de entidades.

```
aws comprehend list-entity-recognizer-summaries
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Reconhecimento personalizado de entidades](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [ListEntityRecognizerSummaries](#) na Referência de AWS CLI Comandos.

list-entity-recognizers

O código de exemplo a seguir mostra como usar `list-entity-recognizers`.

AWS CLI

Para listar todos os reconhecedores de entidades personalizados

O `list-entity-recognizers` exemplo a seguir lista todos os reconhecedores de entidades personalizados criados.

```
aws comprehend list-entity-recognizers
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Reconhecimento personalizado de entidades](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [ListEntityRecognizers](#) na Referência de AWS CLI Comandos.

list-events-detection-jobs

O código de exemplo a seguir mostra como usar `list-events-detection-jobs`.

AWS CLI

Para listar todos os trabalhos de detecção de eventos

O `list-events-detection-jobs` exemplo a seguir lista todos os trabalhos de detecção de eventos assíncronos.

```
aws comprehend list-events-detection-jobs
```

Saída:

```
{
  "EventsDetectionJobPropertiesList": [
    {
      "JobId": "aa9593f9203e84f3ef032ce18EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/aa9593f9203e84f3ef032ce18EXAMPLE",
      "JobName": "events_job_1",
      "JobStatus": "COMPLETED",
      "SubmitTime": "2023-06-12T19:14:57.751000+00:00",
      "EndTime": "2023-06-12T19:21:04.962000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/EventsData/",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/1111222233333-EVENTS-aa9593f9203e84f3ef032ce18EXAMPLE/output/"
      },
      "LanguageCode": "en",
      "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
      "TargetEventTypes": [
        "BANKRUPTCY",
        "EMPLOYMENT",
        "CORPORATE_ACQUISITION",
        "CORPORATE_MERGER",
        "INVESTMENT_GENERAL"
      ]
    },
    {
      "JobId": "4a990a2f7e82adfca6e171135EXAMPLE",
```

```

    "JobArn": "arn:aws:comprehend:us-west-2:1111222233333:events-detection-
job/4a990a2f7e82adfca6e171135EXAMPLE",
    "JobName": "events_job_2",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2023-06-12T19:55:43.702000+00:00",
    "EndTime": "2023-06-12T20:03:49.893000+00:00",
    "InputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-SOURCE-BUCKET/EventsData/",
      "InputFormat": "ONE_DOC_PER_LINE"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/1111222233333-EVENTS-4a990a2f7e82adfca6e171135EXAMPLE/output/"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::1111222233333:role/service-role/
AmazonComprehendServiceRole-example-role",
    "TargetEventTypes": [
      "BANKRUPTCY",
      "EMPLOYMENT",
      "CORPORATE_ACQUISITION",
      "CORPORATE_MERGER",
      "INVESTMENT_GENERAL"
    ]
  }
]
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListEventsDetectionJobs](#) na Referência de AWS CLI Comandos.

list-flywheel-iteration-history

O código de exemplo a seguir mostra como usar `list-flywheel-iteration-history`.

AWS CLI

Para listar todo o histórico de iteração do volante

O `list-flywheel-iteration-history` exemplo a seguir lista todas as iterações de um volante.

```
aws comprehend list-flywheel-iteration-history
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

Saída:

```
{
  "FlywheelIterationPropertiesList": [
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",
      "FlywheelIterationId": "20230619TEXAMPLE",
      "CreationTime": "2023-06-19T04:00:32.594000+00:00",
      "EndTime": "2023-06-19T04:00:49.248000+00:00",
      "Status": "COMPLETED",
      "Message": "FULL_ITERATION: Flywheel iteration performed all functions successfully.",
      "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier/version/1",
      "EvaluatedModelMetrics": {
        "AverageF1Score": 0.7742663922375772,
        "AverageF1Score": 0.9876464664646313,
        "AveragePrecision": 0.9800000253081214,
        "AverageRecall": 0.9445600253081214,
        "AverageAccuracy": 0.9997281665190434
      },
      "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-flywheel/schemaVersion=1/20230619TEXAMPLE/evaluation/20230619TEXAMPLE/"
    },
    {
      "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-2",
      "FlywheelIterationId": "20230616TEXAMPLE",
      "CreationTime": "2023-06-16T21:10:26.385000+00:00",
      "EndTime": "2023-06-16T23:33:16.827000+00:00",
      "Status": "COMPLETED",
      "Message": "FULL_ITERATION: Flywheel iteration performed all functions successfully.",
      "EvaluatedModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/spamvshamclassify/version/1",
```

```

    "EvaluatedModelMetrics": {
      "AverageF1Score": 0.7742663922375772,
      "AverageF1Score": 0.9767700253081214,
      "AveragePrecision": 0.9767700253081214,
      "AverageRecall": 0.9767700253081214,
      "AverageAccuracy": 0.9858281665190434
    },
    "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-
flywheel-2/schemaVersion=1/20230616TEXAMPLE/evaluation/20230616TEXAMPLE/"
  }
]
}

```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [ListFlywheelIterationHistory](#) na Referência de AWS CLI Comandos.

list-flywheels

O código de exemplo a seguir mostra como usar `list-flywheels`.

AWS CLI

Para listar todos os volantes

O `list-flywheels` exemplo a seguir lista todos os volantes criados.

```
aws comprehend list-flywheels
```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [ListFlywheels](#) na Referência de AWS CLI Comandos.

list-key-phrases-detection-jobs

O código de exemplo a seguir mostra como usar `list-key-phrases-detection-jobs`.

AWS CLI

Para listar todos os trabalhos de detecção de frases-chave

O `list-key-phrases-detection-jobs` exemplo a seguir lista todos os trabalhos de detecção de frases-chave assíncronas em andamento e concluídos.

```
aws comprehend list-key-phrases-detection-jobs
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListKeyPhrasesDetectionJobs](#) na Referência de AWS CLI Comandos.

list-pii-entities-detection-jobs

O código de exemplo a seguir mostra como usar `list-pii-entities-detection-jobs`.

AWS CLI

Para listar todos os trabalhos de detecção de entidades pii

O `list-pii-entities-detection-jobs` exemplo a seguir lista todos os trabalhos de detecção assíncrona de pii em andamento e concluídos.

```
aws comprehend list-pii-entities-detection-jobs
```

Saída:

```
{
```

```
"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"
  }
]
```

```
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListPiiEntitiesDetectionJobs](#) na Referência de AWS CLI Comandos.

list-sentiment-detection-jobs

O código de exemplo a seguir mostra como usar `list-sentiment-detection-jobs`.

AWS CLI

Para listar todos os trabalhos de detecção de sentimentos

O `list-sentiment-detection-jobs` exemplo a seguir lista todos os trabalhos de detecção de sentimentos assíncronos em andamento e concluídos.

```
aws comprehend list-sentiment-detection-jobs
```

Saída:

```
{
  "SentimentDetectionJobPropertiesList": [
    {
      "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
      "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
      "JobName": "example-sentiment-detection-job",
      "JobStatus": "IN_PROGRESS",
      "SubmitTime": "2023-06-09T22:42:20.545000+00:00",
      "EndTime": "2023-06-09T22:52:27.416000+00:00",
      "InputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData",
        "InputFormat": "ONE_DOC_PER_LINE"
      },
      "OutputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a11EXAMPLE/output/output.tar.gz"
      },
    }
  ]
}
```

```

        "LanguageCode": "en",
        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    },
    {
        "JobId": "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"
    }
]
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListSentimentDetectionJobs](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para o recurso

O `list-tags-for-resource` exemplo a seguir lista as tags de um recurso do Amazon Comprehend.

```
aws comprehend list-tags-for-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1
```

Saída:

```
{  
  "ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1",  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Finance"  
    },  
    {  
      "Key": "location",  
      "Value": "Seattle"  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-targeted-sentiment-detection-jobs

O código de exemplo a seguir mostra como usar `list-targeted-sentiment-detection-jobs`.

AWS CLI

Para listar todos os trabalhos direcionados de detecção de sentimentos

O `list-targeted-sentiment-detection-jobs` exemplo a seguir lista todos os trabalhos assíncronos de detecção de sentimentos direcionados em andamento e concluídos.

```
aws comprehend list-targeted-sentiment-detection-jobs
```

Saída:

```
{
```

```
"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"
  }
]
}
```


Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListTargetedSentimentDetectionJobs](#) na Referência de AWS CLI Comandos.

list-topics-detection-jobs

O código de exemplo a seguir mostra como usar `list-topics-detection-jobs`.

AWS CLI

Para listar todos os trabalhos de detecção de tópicos

O exemplo de `list-topics-detection-jobs` a seguir lista todos os trabalhos de detecção de tópicos assíncronos em andamento e concluídos.

```
aws comprehend list-topics-detection-jobs
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [ListTopicsDetectionJobs](#) na Referência de AWS CLI Comandos.

put-resource-policy

O código de exemplo a seguir mostra como usar `put-resource-policy`.

AWS CLI

Para anexar uma política baseada em recursos

O `put-resource-policy` exemplo a seguir anexa uma política baseada em recursos a um modelo para que ela possa ser importada por outra conta. AWS A política é anexada ao modelo na conta 111122223333 e permite que a conta 444455556666 importe o modelo.

```
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"]}]}']'
```

Saída:

```
{  
  "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"  
}
```

Para obter mais informações, consulte [Cópia de modelos personalizados entre AWS contas](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [PutResourcePolicy](#) na Referência de AWS CLI Comandos.

start-document-classification-job

O código de exemplo a seguir mostra como usar `start-document-classification-job`.

AWS CLI

Para iniciar um trabalho de classificação de documento

O exemplo de `start-document-classification-job` a seguir inicia um trabalho de classificação de documentos com um modelo personalizado em todos os arquivos no endereço especificado pela tag `--input-data-config`. Neste exemplo, o bucket do S3 de entrada contém `SampleSMStext1.txt`, `SampleSMStext2.txt` e `SampleSMStext3.txt`. O modelo foi previamente treinado em classificações de documentos de spam e não spam, ou mensagens “falsas”. SMS Quando o trabalho é concluído, `output.tar.gz` é colocado no local especificado pela tag `--output-data-config`. `output.tar.gz` contém `predictions.jsonl`, que lista a classificação de cada documento. A saída Json é impressa em uma linha por arquivo, mas foi formatada aqui para facilitar a leitura.

```
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
```

Conteúdo de `SampleSMStext1.txt`:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Conteúdo de `SampleSMStext2.txt`:

```
"Hi, when do you want me to pick you up from practice?"
```

Conteúdo de `SampleSMStext3.txt`:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Saída:

```
{  
  "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
```

```

    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
    job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
  }

```

Conteúdo de `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}]}

```

Para obter mais informações, consulte [Classificação personalizada](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartDocumentClassificationJob](#) na Referência de AWS CLI Comandos.

start-dominant-language-detection-job

O código de exemplo a seguir mostra como usar `start-dominant-language-detection-job`.

AWS CLI

Para iniciar um trabalho assíncrono de detecção de linguagem

O `start-dominant-language-detection-job` exemplo a seguir inicia um trabalho assíncrono de detecção de idioma para todos os arquivos localizados no endereço especificado pela tag `--input-data-config`. O bucket do S3 neste exemplo contém `Sampletext1.txt`. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta contém `output.txt` o idioma dominante de cada um dos arquivos de texto, bem como a pontuação de confiança do modelo pré-treinado para cada previsão.

```

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
```

Conteúdo do 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."
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-detection-job/123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

Conteúdo de output.txt:

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score": 0.9913753867149353}], "Line": 0}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartDominantLanguageDetectionJob](#) na Referência de AWS CLI Comandos.

start-entities-detection-job

O código de exemplo a seguir mostra como usar start-entities-detection-job.

AWS CLI

Exemplo 1: Para iniciar um trabalho de detecção de entidade padrão usando o modelo pré-treinado

O start-entities-detection-job exemplo a seguir inicia um trabalho de detecção de entidades assíncronas para todos os arquivos localizados no endereço

especificado pela tag. `--input-data-config` O bucket do S3 neste exemplo contém `Sampletext1.txt`, `Sampletext2.txt`, e `Sampletext3.txt`. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta `output.txt` contém a lista de todas as entidades nomeadas detectadas em cada arquivo de texto, bem como a pontuação de confiança do modelo pré-treinado para cada previsão. A saída Json é impressa em uma linha por arquivo de entrada, mas é formatada aqui para facilitar a leitura.

```
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
```

Conteúdo de `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."
```

Conteúdo de `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. "
```

Conteúdo de `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."
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",
```

```
"JobStatus": "SUBMITTED"  
}
```

Conteúdo de output.txt com recuos de linha para facilitar a leitura:

```
{  
  "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
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

Exemplo 2: Para iniciar um trabalho personalizado de detecção de entidades

O `start-entities-detection-job` exemplo a seguir inicia um trabalho assíncrono de detecção de entidades personalizadas para todos os arquivos localizados no endereço especificado pela tag `--input-data-config`. Neste exemplo, o bucket do S3 neste exemplo contém `SampleFeedback1.txt`, `SampleFeedback2.txt`, e `SampleFeedback3.txt`. O modelo de reconhecimento de entidades foi treinado em feedbacks de suporte ao cliente para reconhecer nomes de dispositivos. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta contém `output.txt`, que lista todas as entidades nomeadas detectadas em cada arquivo de texto, bem como a pontuação de confiança do modelo pré-treinado para cada previsão. A saída Json é impressa em uma linha por arquivo, mas foi formatada aqui para facilitar a leitura.

```
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"
```

Conteúdo de `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!"
```

Conteúdo de `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!"
```

Conteúdo de `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!"
```

Saída:

```
{
  "JobId": "019ea9edac758806850fa8a79ff83021",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-job/019ea9edac758806850fa8a79ff83021",
  "JobStatus": "SUBMITTED"
}
```

Conteúdo de output.txt com recuos de linha para facilitar a leitura:

```
{
  "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
}

```

Para obter mais informações, consulte [Reconhecimento personalizado de entidades](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [StartEntitiesDetectionJob](#) na Referência de AWS CLI Comandos.

start-events-detection-job

O código de exemplo a seguir mostra como usar `start-events-detection-job`.

AWS CLI

Para iniciar um trabalho de detecção de eventos assíncronos

O `start-events-detection-job` exemplo a seguir inicia um trabalho de detecção de eventos assíncronos para todos os arquivos localizados no endereço especificado pela tag `--input-data-config`. Os possíveis tipos de eventos-alvo incluem BANKRUPTCY, EMPLOYMENT, CORPORATE_ACQUISITION, INVESTMENT_GENERAL, CORPORATE_MERGER, IPO, RIGHTS_ISSUE, SECOND_STOCK_SPLIT e. O bucket do S3 neste exemplo contém `SampleText1.txt`, `SampleText2.txt`, e `SampleText3.txt`. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta contém `SampleText1.txt.out`, `SampleText2.txt.out`, `SampleText3.txt.out` e. A JSON saída é impressa em uma linha por arquivo, mas é formatada aqui para facilitar a leitura.

```

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_GENERER"
```

Conteúdo de 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."
```

Conteúdo de SampleText2.txt:

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars, surprising and exciting the shareholders."
```

Conteúdo de SampleText3.txt:

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for bankruptcy."
```

Saída:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Conteúdo de SampleText1.txt.out com recuos de linha para facilitar a leitura:

```
{
  "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
}
```

Conteúdo de 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
}

```

Conteúdo de 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
      }
    ]
  },
  {
    "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
}

```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartEventsDetectionJob](#) na Referência de AWS CLI Comandos.

start-flywheel-iteration

O código de exemplo a seguir mostra como usar `start-flywheel-iteration`.

AWS CLI

Para iniciar uma iteração do volante

O `start-flywheel-iteration` exemplo a seguir inicia uma iteração do volante. Essa operação usa qualquer novo conjunto de dados no volante para treinar uma nova versão do modelo.

```

aws comprehend start-flywheel-iteration \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel

```

Saída:

```

{
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",

```

```
"FlywheelIterationId": "12345123TEXAMPLE"
}
```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [StartFlywheelIteration](#) na Referência de AWS CLI Comandos.

start-key-phrases-detection-job

O código de exemplo a seguir mostra como usar `start-key-phrases-detection-job`.

AWS CLI

Para iniciar um trabalho de detecção de frases-chave

O `start-key-phrases-detection-job` exemplo a seguir inicia um trabalho assíncrono de detecção de frases-chave para todos os arquivos localizados no endereço especificado pela tag. `--input-data-config` O bucket do S3 neste exemplo contém `Sampletext1.txt`, `Sampletext2.txt`, e `Sampletext3.txt`. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta contém o arquivo `output.txt` que contém todas as frases-chave detectadas em cada arquivo de texto e a pontuação de confiança do modelo pré-treinado para cada previsão. A saída Json é impressa em uma linha por arquivo, mas foi formatada aqui para facilitar a leitura.

```
aws comprehend start-key-phrases-detection-job \
  --job-name keyphrasesanalysisitest1 \
  --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
```

Conteúdo de `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."
```

Conteúdo de `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. "
```

Conteúdo de 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."
```

Saída:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Conteúdo de output.txt com recuos de linha para facilitar a leitura:

```
{
  "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
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartKeyPhrasesDetectionJob](#) na Referência de AWS CLI Comandos.

start-pii-entities-detection-job

O código de exemplo a seguir mostra como usar `start-pii-entities-detection-job`.

AWS CLI

Para iniciar um trabalho de detecção assíncrona PII

O `start-pii-entities-detection-job` exemplo a seguir inicia um trabalho assíncrono de detecção de entidades de informações de identificação pessoal (PII) para todos os arquivos localizados no endereço especificado pela tag. `--input-data-config` O bucket do S3 neste exemplo contém `Sampletext1.txt`, `Sampletext2.txt`, e `Sampletext3.txt`. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta contém `SampleText1.txt.out`, `SampleText2.txt.out`, e lista `SampleText3.txt.out` as entidades nomeadas em cada arquivo de texto. A saída Json é impressa em uma linha por arquivo, mas foi formatada aqui para facilitar a leitura.

```
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
```

Conteúdo de `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."
```

Conteúdo de `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. "
```

Conteúdo de 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."
```

Saída:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Conteúdo de SampleText1.txt.out com recuos de linha para facilitar a leitura:

```
{
  "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  
}
```

Conteúdo de SampleText2.txt.out com recuos de linha para facilitar a leitura:

```
{  
  "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  
}
```

Conteúdo de SampleText3.txt.out com recuos de linha para facilitar a leitura:

```
{
  "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
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartPiiEntitiesDetectionJob](#) na Referência de AWS CLI Comandos.

start-sentiment-detection-job

O código de exemplo a seguir mostra como usar `start-sentiment-detection-job`.

AWS CLI

Para iniciar um trabalho assíncrono de análise de sentimentos

O `start-sentiment-detection-job` exemplo a seguir inicia um trabalho assíncrono de detecção de análise de sentimentos para todos os arquivos localizados no endereço especificado pela tag. `--input-data-config` A pasta de bucket do S3 neste exemplo contém `SampleMovieReview1.txt`, `SampleMovieReview2.txt`, e `SampleMovieReview3.txt`. Quando o trabalho é concluído, a pasta `output`, é colocada no local especificado pela `--output-data-config` tag. A pasta contém o arquivo `output.txt`, que contém os sentimentos predominantes para cada arquivo de texto e a pontuação de confiança do modelo pré-treinado para cada previsão. A saída Json é impressa em uma linha por arquivo, mas foi formatada aqui para facilitar a leitura.

```
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
```

Conteúdo de `SampleMovieReview1.txt`:

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

Conteúdo de `SampleMovieReview2.txt`:

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I highly recommend this movie."
```

Conteúdo de `SampleMovieReview3.txt`:

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly stolen from other modern directors."
```

Saída:

```
{
```

```
"JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
"JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/0b5001e25f62ebb40631a9a1a7fde7b3",
"JobStatus": "SUBMITTED"
}
```

Conteúdo de output.txt com linha de recuos para facilitar a leitura:

```
{
  "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
  }
}
}
```


Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartSentimentDetectionJob](#) na Referência de AWS CLI Comandos.

start-targeted-sentiment-detection-job

O código de exemplo a seguir mostra como usar `start-targeted-sentiment-detection-job`.

AWS CLI

Para iniciar um trabalho assíncrono de análise de sentimentos direcionados

O `start-targeted-sentiment-detection-job` exemplo a seguir inicia um trabalho assíncrono de detecção de análise de sentimentos direcionada para todos os arquivos localizados no endereço especificado pela tag. `--input-data-config` A pasta de bucket do S3 neste exemplo contém `SampleMovieReview1.txt`, `SampleMovieReview2.txt`, e `SampleMovieReview3.txt`. Quando o trabalho é concluído, `output.tar.gz` é colocado no local especificado pela `--output-data-config` etiqueta. `output.tar.gz` contém os arquivos `SampleMovieReview1.txt.out`, `SampleMovieReview2.txt.out`, e `SampleMovieReview3.txt.out`, cada um contendo todas as entidades nomeadas e sentimentos associados a um único arquivo de texto de entrada.

```
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
```

Conteúdo de `SampleMovieReview1.txt`:

```
"The film, AnyMovie, is fairly predictable and just okay."
```

Conteúdo de `SampleMovieReview2.txt`:

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I  
highly recommend this movie."
```

Conteúdo de SampleMovieReview3.txt:

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly
stolen from other modern directors."
```

Saída:

```
{
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",
  "JobStatus": "SUBMITTED"
}
```

Conteúdo de SampleMovieReview1.txt.out com recuos de linha para facilitar a leitura:

```
{
  "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
}
```

Conteúdo dos recuos de SampleMovieReview2.txt.out linha para facilitar a leitura:

```
{
  "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  
}
```

Conteúdo de SampleMovieReview3.txt.out com recuos de linha para facilitar a leitura:

```
{  
  "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": 0.960058,  
        "Positive": 0.960058  
      }  
    }  
  }  
]
```

```
        "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
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartTargetedSentimentDetectionJob](#) na Referência de AWS CLI Comandos.

start-topics-detection-job

O código de exemplo a seguir mostra como usar `start-topics-detection-job`.

AWS CLI

Para iniciar um trabalho de análise de detecção de tópicos

O exemplo de `start-topics-detection-job` a seguir inicia um trabalho assíncrono de detecção de tópicos para todos os arquivos localizados no endereço especificado pela tag `--input-data-config`. Quando o trabalho é concluído, a pasta, `output`, é colocada no local especificado pela tag `--output-data-config`. `output` contém `topic-terms.csv` e `doc-topics.csv`. O primeiro arquivo de saída, `topic-terms.csv`, é uma lista de tópicos na coleção. Para cada tópico, a lista inclui por padrão os principais termos por tópico de acordo com seu peso. O segundo arquivo, `doc-topics.csv`, lista os documentos associados a um tópico e a proporção do documento relacionada ao tópico.

```
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
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

Para obter mais informações, consulte [Modelagem de tópicos](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StartTopicsDetectionJob](#) na Referência de AWS CLI Comandos.

stop-dominant-language-detection-job

O código de exemplo a seguir mostra como usar stop-dominant-language-detection-job.

AWS CLI

Para interromper um trabalho assíncrono de detecção de linguagem dominante

O stop-dominant-language-detection-job exemplo a seguir interrompe um trabalho de detecção assíncrona de linguagem dominante em andamento. Se o estado atual do trabalho for, IN_PROGRESS o trabalho será marcado para rescisão e colocado no STOP_REQUESTED estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no COMPLETED estado.

```
aws comprehend stop-dominant-language-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
```

```
"JobStatus": "STOP_REQUESTED"
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopDominantLanguageDetectionJob](#) na Referência de AWS CLI Comandos.

stop-entities-detection-job

O código de exemplo a seguir mostra como usar `stop-entities-detection-job`.

AWS CLI

Para interromper um trabalho de detecção de entidades assíncronas

O `stop-entities-detection-job` exemplo a seguir interrompe um trabalho de detecção de entidades assíncronas em andamento. Se o estado atual do trabalho for, `IN_PROGRESS` o trabalho será marcado para rescisão e colocado no `STOP_REQUESTED` estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no `COMPLETED` estado.

```
aws comprehend stop-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopEntitiesDetectionJob](#) na Referência de AWS CLI Comandos.

stop-events-detection-job

O código de exemplo a seguir mostra como usar `stop-events-detection-job`.

AWS CLI

Para interromper um trabalho de detecção de eventos assíncronos

O `stop-events-detection-job` exemplo a seguir interrompe um trabalho de detecção de eventos assíncronos em andamento. Se o estado atual do trabalho for, `IN_PROGRESS` o trabalho será marcado para rescisão e colocado no `STOP_REQUESTED` estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no `COMPLETED` estado.

```
aws comprehend stop-events-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopEventsDetectionJob](#) na Referência de AWS CLI Comandos.

stop-key-phrases-detection-job

O código de exemplo a seguir mostra como usar `stop-key-phrases-detection-job`.

AWS CLI

Para interromper um trabalho assíncrono de detecção de frases-chave

O `stop-key-phrases-detection-job` exemplo a seguir interrompe um trabalho de detecção de frases-chave assíncrono em andamento. Se o estado atual do trabalho for, `IN_PROGRESS` o trabalho será marcado para rescisão e colocado no `STOP_REQUESTED` estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no `COMPLETED` estado.

```
aws comprehend stop-key-phrases-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopKeyPhrasesDetectionJob](#) na Referência de AWS CLI Comandos.

stop-pii-entities-detection-job

O código de exemplo a seguir mostra como usar `stop-pii-entities-detection-job`.

AWS CLI

Para interromper um trabalho assíncrono de detecção de entidades pii

O `stop-pii-entities-detection-job` exemplo a seguir interrompe um trabalho de detecção de entidades pii assíncronas e em andamento. Se o estado atual do trabalho for, `IN_PROGRESS` o trabalho será marcado para rescisão e colocado no `STOP_REQUESTED` estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no `COMPLETED` estado.

```
aws comprehend stop-pii-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopPiiEntitiesDetectionJob](#) na Referência de AWS CLI Comandos.

stop-sentiment-detection-job

O código de exemplo a seguir mostra como usar `stop-sentiment-detection-job`.

AWS CLI

Para interromper um trabalho assíncrono de detecção de sentimentos

O `stop-sentiment-detection-job` exemplo a seguir interrompe um trabalho assíncrono de detecção de sentimentos em andamento. Se o estado atual do trabalho for, `IN_PROGRESS` o trabalho será marcado para rescisão e colocado no `STOP_REQUESTED` estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no `COMPLETED` estado.

```
aws comprehend stop-sentiment-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopSentimentDetectionJob](#) na Referência de AWS CLI Comandos.

stop-targeted-sentiment-detection-job

O código de exemplo a seguir mostra como usar `stop-targeted-sentiment-detection-job`.

AWS CLI

Para interromper um trabalho assíncrono de detecção de sentimentos direcionados

O `stop-targeted-sentiment-detection-job` exemplo a seguir interrompe um trabalho assíncrono de detecção de sentimentos direcionados em andamento. Se o estado atual do trabalho for, `IN_PROGRESS` o trabalho será marcado para rescisão e colocado no

STOP_REQUESTED estado. Se o trabalho for concluído antes de ser interrompido, ele será colocado no COMPLETED estado.

```
aws comprehend stop-targeted-sentiment-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Saída:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Para obter mais informações, consulte [Análise assíncrona para insights do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopTargetedSentimentDetectionJob](#) na Referência de AWS CLI Comandos.

stop-training-document-classifier

O código de exemplo a seguir mostra como usar stop-training-document-classifier.

AWS CLI

Para interromper o treinamento de um modelo de classificador de documentos

O stop-training-document-classifier exemplo a seguir interrompe o treinamento de um modelo de classificador de documentos durante o andamento.

```
aws comprehend stop-training-document-classifier  
--document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e gerenciamento de modelos personalizados](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopTrainingDocumentClassifier](#) na Referência de AWS CLI Comandos.

stop-training-entity-recognizer

O código de exemplo a seguir mostra como usar `stop-training-entity-recognizer`.

AWS CLI

Para interromper o treinamento de um modelo reconhecedor de entidades

O `stop-training-entity-recognizer` exemplo a seguir interrompe o treinamento de um modelo de reconhecimento de entidades enquanto está em andamento.

```
aws comprehend stop-training-entity-recognizer
  --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/examplerecognizer1"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e gerenciamento de modelos personalizados](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [StopTrainingEntityRecognizer](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Exemplo 1: Para marcar um recurso

O `tag-resource` exemplo a seguir adiciona uma única tag a um recurso do 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
```

Esse comando não tem saída.

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do desenvolvedor do Amazon Comprehend.

Exemplo 2: Para adicionar várias tags a um recurso

O `tag-resource` exemplo a seguir adiciona várias tags a um recurso do 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
```

Esse comando não tem saída.

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Exemplo 1: Para remover uma única tag de um recurso

O `untag-resource` exemplo a seguir remove uma única tag de um recurso do Amazon Comprehend.

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1  
  --tag-keys Location
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do desenvolvedor do Amazon Comprehend.

Exemplo 2: Para remover várias tags de um recurso

O `untag-resource` exemplo a seguir remove várias tags de um recurso do Amazon Comprehend.

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1  
  --tag-keys Location
```

```
--resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1  
--tag-keys Location Department
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-endpoint

O código de exemplo a seguir mostra como usar update-endpoint.

AWS CLI

Exemplo 1: Para atualizar as unidades de inferência de um endpoint

O update-endpoint exemplo a seguir atualiza as informações sobre um endpoint. Neste exemplo, o número de unidades de inferência é aumentado.

```
aws comprehend update-endpoint \  
--endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint  
--desired-inference-units 2
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

Exemplo 2: Para atualizar o modelo ativo de um endpoint

O update-endpoint exemplo a seguir atualiza as informações sobre um endpoint. Neste exemplo, o modelo ativo é alterado.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar endpoints do Amazon Comprehend](#) no Guia do desenvolvedor do Amazon Comprehend.

- Para API obter detalhes, consulte [UpdateEndpoint](#) na Referência de AWS CLI Comandos.

update-flywheel

O código de exemplo a seguir mostra como usar `update-flywheel`.

AWS CLI

Para atualizar a configuração de um volante

O `update-flywheel` exemplo a seguir atualiza a configuração do volante. Neste exemplo, o modelo ativo do volante é atualizado.

```
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
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte a [visão geral do Flywheel](#) no Amazon Comprehend Developer Guide.

- Para API obter detalhes, consulte [UpdateFlywheel](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Comprehend Medical usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Comprehend Medical.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-entities-detection-v2-job

O código de exemplo a seguir mostra como usar `describe-entities-detection-v2-job`.

AWS CLI

Para descrever um trabalho de detecção de entidades

O `describe-entities-detection-v2-job` exemplo a seguir exibe as propriedades associadas a um trabalho assíncrono de detecção de entidades.

```
aws comprehendmedical describe-entities-detection-v2-job \
  --job-id "ab9887877365fe70299089371c043b96"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [DescribeEntitiesDetectionV2Job na Referência](#) de AWS CLI Comandos.

describe-icd10-cm-inference-job

O código de exemplo a seguir mostra como usar `describe-icd10-cm-inference-job`.

AWS CLI

Para descrever um trabalho de ICD inferência de -10 CM

O `describe-icd10-cm-inference-job` exemplo a seguir descreve as propriedades do trabalho de inferência solicitado com o `job-id` especificado.

```
aws comprehendmedical describe-icd10-cm-inference-job \
  --job-id "5780034166536cdb52ffa3295a1b00a7"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [DescribeIcd10 CmInferenceJob](#) na Referência de AWS CLI Comandos.

describe-phi-detection-job

O código de exemplo a seguir mostra como usar `describe-phi-detection-job`.

AWS CLI

Para descrever um trabalho PHI de detecção

O `describe-phi-detection-job` exemplo a seguir exibe as propriedades associadas a uma tarefa assíncrona de detecção de informações de saúde protegidas (PHI).

```
aws comprehendmedical describe-phi-detection-job \
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [DescribePhiDetectionJob](#) na Referência de AWS CLI Comandos.

describe-rx-norm-inference-job

O código de exemplo a seguir mostra como usar `describe-rx-norm-inference-job`.

AWS CLI

Para descrever um trabalho de RxNorm inferência

O `describe-rx-norm-inference-job` exemplo a seguir descreve as propriedades do trabalho de inferência solicitado com o `job-id` especificado.

```
aws comprehendmedical describe-rx-norm-inference-job \
  --job-id "eg8199877365fc70299089371c043b96"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [DescribeRxNormInferenceJob](#) na Referência de AWS CLI Comandos.

describe-snomedct-inference-job

O código de exemplo a seguir mostra como usar `describe-snomedct-inference-job`.

AWS CLI

Para descrever um trabalho de inferência de SNOMED tomografia computadorizada

O `describe-snomedct-inference-job` exemplo a seguir descreve as propriedades do trabalho de inferência solicitado com o `job-id` especificado.

```
aws comprehendmedical describe-snomedct-inference-job \  
  --job-id "2630034166536cdb52ffa3295a1b00a7"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [DescribeSnomedctInferenceJob](#) na Referência de AWS CLI Comandos.

detect-entities-v2

O código de exemplo a seguir mostra como usar `detect-entities-v2`.

AWS CLI

Exemplo 1: Para detectar entidades diretamente do texto

O `detect-entities-v2` exemplo a seguir mostra as entidades detectadas e as rotula de acordo com o tipo, diretamente do texto de entrada.

```
aws comprehendmedical detect-entities-v2 \  
  --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and  
  leg, slightly itchy."
```

Saída:

```
{  
  "Id": 0,  
  "BeginOffset": 38,  
  "EndOffset": 47,  
  "Score": 0.9942955374717712,  
  "Text": "Clonidine",  
  "Category": "MEDICATION",  
  "Type": "GENERIC_NAME",  
  "Traits": []  
}
```

Para obter mais informações, consulte [Detect Entities versão 2](#) no Amazon Comprehend Medical Developer Guide.

Exemplo 2: Para detectar entidades a partir de um caminho de arquivo

O `detect-entities-v2` exemplo a seguir mostra as entidades detectadas e as rotula de acordo com o tipo de um caminho de arquivo.

```
aws comprehendmedical detect-entities-v2 \  
  --text file://medical_entities.txt
```

Conteúdo de `medical_entities.txt`:

```
{
```

```
"Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,
slightly itchy."
}
```

Saída:

```
{
  "Id": 0,
  "BeginOffset": 38,
  "EndOffset": 47,
  "Score": 0.9942955374717712,
  "Text": "Clonidine",
  "Category": "MEDICATION",
  "Type": "GENERIC_NAME",
  "Traits": []
}
```

Para obter mais informações, consulte [Detect Entities versão 2](#) no Amazon Comprehend Medical Developer Guide.

- Para API obter detalhes, consulte [DetectEntitiesV2](#) na Referência de AWS CLI Comandos.

detect-phi

O código de exemplo a seguir mostra como usar detect-phi.

AWS CLI

Exemplo 1: Para detectar informações de saúde protegidas (PHI) diretamente do texto

O detect-phi exemplo a seguir exibe as entidades de informações de saúde protegidas (PHI) detectadas diretamente do texto de entrada.

```
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."
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Detect PHI](#) no Amazon Comprehend Medical Developer Guide.

Exemplo 2: Para detectar informações de saúde protegidas (PHI) diretamente de um caminho de arquivo

O `detect-phi` exemplo a seguir mostra as entidades de informações de saúde protegidas (PHI) detectadas em um caminho de arquivo.

```
aws comprehendmedical detect-phi \  
--text file://phi.txt
```

Conteúdo de 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."
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [Detect PHI](#) no Amazon Comprehend Medical Developer Guide.

- Para API obter detalhes, consulte [DetectPhina Referência de AWS CLI Comandos](#).

infer-icd10-cm

O código de exemplo a seguir mostra como usar `infer-icd10-cm`.

AWS CLI

Exemplo 1: Para detectar entidades de condições médicas e vincular à Ontologia ICD -10-CM diretamente do texto

O `infer-icd10-cm` exemplo a seguir rotula as entidades de condições médicas detectadas e vincula essas entidades aos códigos na edição de 2019 da Classificação Internacional de Doenças e Modificação Clínica (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."

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [Infer ICD1 0-CM](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

Exemplo 2: Para detectar entidades de condições médicas e vincular à ontologia ICD -10-CM a partir de um caminho de arquivo

O `infer-icd-10-cm` exemplo a seguir rotula as entidades de condições médicas detectadas e vincula essas entidades aos códigos na edição de 2019 da Classificação Internacional de Doenças e Modificação Clínica (ICD-10-CM).

```
aws comprehendmedical infer-icd10-cm \  
  --text file://icd10cm.txt
```

Conteúdo de `icd10cm.txt`:

```
{  
  "The patient complains of abdominal pain, has a long-standing history of  
  diabetes treated with Micronase daily."  
}
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [ICD1Infer-0-CM](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [InferIcd10Cm](#) na Referência de AWS CLI Comandos.

infer-rx-norm

O código de exemplo a seguir mostra como usar `infer-rx-norm`.

AWS CLI

Exemplo 1: Para detectar entidades medicamentosas e vinculá-las RxNorm diretamente do texto

O `infer-rx-norm` exemplo a seguir mostra e rotula as entidades medicamentosas detectadas e vincula essas entidades aos identificadores de conceito (RxCUI) do banco de dados da 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."
```

Saída:

```
{
  "Entities": [
    {
      "Id": 0,
      "Text": "Levothyroxine",
      "Category": "MEDICATION",
      "Type": "GENERIC_NAME",
      "Score": 0.9996285438537598,
      "BeginOffset": 23,
      "EndOffset": 36,
      "Attributes": [
        {
          "Type": "DOSAGE",
          "Score": 0.9892290830612183,
          "RelationshipScore": 0.9997978806495667,
          "Id": 1,
          "BeginOffset": 37,
          "EndOffset": 51,
          "Text": "125 micrograms",
          "Traits": []
        },
        {
          "Type": "ROUTE_OR_MODE",
          "Score": 0.9988924860954285,
          "RelationshipScore": 0.998291552066803,
          "Id": 2,
          "BeginOffset": 52,
          "EndOffset": 56,
          "Text": "p.o.",
          "Traits": []
        },
        {
          "Type": "FREQUENCY",
          "Score": 0.9953463673591614,
          "RelationshipScore": 0.9999889135360718,
          "Id": 3,
          "BeginOffset": 57,
          "EndOffset": 67,

```

```

        "Text": "once daily",
        "Traits": []
    }
],
"Traits": [],
"RxNormConcepts": [
    {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
        "Code": "966224",
        "Score": 0.9912070631980896
    },
    {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
        "Code": "966405",
        "Score": 0.8698278665542603
    },
    {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
        "Code": "966191",
        "Score": 0.7448257803916931
    },
    {
        "Description": "levothyroxine",
        "Code": "10582",
        "Score": 0.7050482630729675
    },
    {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxy]",
        "Code": "966190",
        "Score": 0.6921631693840027
    }
]
},
{
    "Id": 4,
    "Text": "Synthroid",
    "Category": "MEDICATION",
    "Type": "BRAND_NAME",
    "Score": 0.9946461319923401,
    "BeginOffset": 86,
    "EndOffset": 95,
    "Attributes": [],

```

```
    "Traits": [
      {
        "Name": "NEGATION",
        "Score": 0.5167351961135864
      }
    ],
    "RxNormConcepts": [
      {
        "Description": "Synthroid",
        "Code": "224920",
        "Score": 0.9462039470672607
      },
      {
        "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
        "Code": "966282",
        "Score": 0.8309829235076904
      },
      {
        "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
        "Code": "966191",
        "Score": 0.4945160448551178
      },
      {
        "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
[Synthroid]",
        "Code": "966247",
        "Score": 0.3674522042274475
      },
      {
        "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
[Synthroid]",
        "Code": "966158",
        "Score": 0.2588822841644287
      }
    ]
  }
],
  "ModelVersion": "0.0.0"
}
```

Para obter mais informações, consulte [Inferir RxNorm no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

Exemplo 2: Para detectar entidades de medicamentos e vinculá-las a RxNorm partir de um caminho de arquivo.

O `infer-rx-norm` exemplo a seguir mostra e rotula as entidades medicamentosas detectadas e vincula essas entidades aos identificadores de conceito (RxCUI) do banco de dados da National Library of Medicine RxNorm .

```
aws comprehendmedical infer-rx-norm \  
  --text file://rxnorm.txt
```

Conteúdo de `rxnorm.txt`:

```
{  
  "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies  
  taking Synthroid."  
}
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [Inferir RxNorm no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [InferRxNorm](#) na Referência de AWS CLI Comandos.

infer-snomedct

O código de exemplo a seguir mostra como usar `infer-snomedct`.

AWS CLI

Exemplo: Para detectar entidades e vincular à Ontologia SNOMED CT diretamente do texto

O `infer-snomedct` exemplo a seguir mostra como detectar entidades médicas e vinculá-las aos conceitos da versão 2021-03 da Nomenclatura Sistematizada da Medicina, Termos Clínicos (CT). SNOMED

```

aws comprehendmedical infer-snomedct \
  --text "The patient complains of abdominal pain, has a long-standing history of
diabetes treated with Micronase daily."

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [Inferir SNOMEDCT no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [InferSnomedct](#) na Referência de AWS CLI Comandos.

list-entities-detection-v2-jobs

O código de exemplo a seguir mostra como usar `list-entities-detection-v2-jobs`.

AWS CLI

Para listar trabalhos de detecção de entidades

O `list-entities-detection-v2-jobs` exemplo a seguir lista os trabalhos atuais de detecção assíncrona.

```
aws comprehendmedical list-entities-detection-v2-jobs
```

Saída:

```

{
  "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"
}
]
}

```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [ListEntitiesDetectionV2Jobs na Referência](#) de AWS CLI Comandos.

list-icd10-cm-inference-jobs

O código de exemplo a seguir mostra como usar `list-icd10-cm-inference-jobs`.

AWS CLI

Para listar todos os trabalhos atuais de ICD inferência de -10-CM

O exemplo a seguir mostra como a `list-icd10-cm-inference-jobs` operação retorna uma lista dos trabalhos de inferência em lote assíncronos atuais de ICD -10 CM.

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

Saída:

```

{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [ListJobProperties](#) na Referência de AWS CLI Comandos.

list-phi-detection-jobs

O código de exemplo a seguir mostra como usar `list-phi-detection-jobs`.

AWS CLI

Para listar trabalhos de detecção de informações de saúde protegidas (PHI)

O `list-phi-detection-jobs` exemplo a seguir lista os trabalhos atuais de detecção de informações de saúde protegidas (PHI)

```
aws comprehendmedical list-phi-detection-jobs
```

Saída:

```

{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [ListPhiDetectionJobs](#) na Referência de AWS CLI Comandos.

list-rx-norm-inference-jobs

O código de exemplo a seguir mostra como usar `list-rx-norm-inference-jobs`.

AWS CLI

Para listar todos os trabalhos atuais de inferência do Rx-Norm

O exemplo a seguir mostra como `list-rx-norm-inference-jobs` retorna uma lista dos trabalhos de inferência em lote assíncronos atuais do Rx-Norm.

```
aws comprehendmedical list-rx-norm-inference-jobs
```

Saída:

```
{
```

```

"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"
  }
]
}

```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [ListRxNormInferenceJobs](#) na Referência de AWS CLI Comandos.

list-snomedct-inference-jobs

O código de exemplo a seguir mostra como usar `list-snomedct-inference-jobs`.

AWS CLI

Para listar todos os trabalhos de inferência de SNOMED CT

O exemplo a seguir mostra como a `list-snomedct-inference-jobs` operação retorna uma lista dos trabalhos de inferência em lote de SNOMED CT assíncrona atuais.

```
aws comprehendmedical list-snomedct-inference-jobs
```

Saída:


```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [ListSnomedctInferenceJobs](#) na Referência de AWS CLI Comandos.

start-entities-detection-v2-job

O código de exemplo a seguir mostra como usar `start-entities-detection-v2-job`.

AWS CLI

Para iniciar um trabalho de detecção de entidades

O `start-entities-detection-v2-job` exemplo a seguir inicia um trabalho assíncrono de detecção de entidades.

```
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
```

Saída:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StartEntitiesDetectionV2Job na Referência](#) de AWS CLI Comandos.

start-icd10-cm-inference-job

O código de exemplo a seguir mostra como usar `start-icd10-cm-inference-job`.

AWS CLI

Para iniciar um trabalho de ICD inferência de -10 CM

O `start-icd10-cm-inference-job` exemplo a seguir inicia um trabalho de ICD análise em lote de inferência de -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
```

Saída:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96"  
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StartIcd10 CmInferenceJob](#) na Referência de AWS CLI Comandos.

start-phi-detection-job

O código de exemplo a seguir mostra como usar start-phi-detection-job.

AWS CLI

Para iniciar um trabalho PHI de detecção

O start-phi-detection-job exemplo a seguir inicia um trabalho assíncrono de detecção de PHI entidades.

```
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
```

Saída:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StartPhiDetectionJob](#) na Referência de AWS CLI Comandos.

start-rx-norm-inference-job

O código de exemplo a seguir mostra como usar start-rx-norm-inference-job.

AWS CLI

Para iniciar um trabalho de RxNorm inferência

O `start-rx-norm-inference-job` exemplo a seguir inicia um trabalho de análise em lote de RxNorm inferência.

```
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
```

Saída:

```
{  
  "JobId": "eg8199877365fc70299089371c043b96"  
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia do desenvolvedor do Amazon Comprehend Medical](#).

- Para API obter detalhes, consulte [StartRxNormInferenceJob](#) na Referência de AWS CLI Comandos.

start-snomedct-inference-job

O código de exemplo a seguir mostra como usar `start-snomedct-inference-job`.

AWS CLI

Para iniciar um trabalho de inferência de SNOMED tomografia computadorizada

O `start-snomedct-inference-job` exemplo a seguir inicia um trabalho de análise em lote de inferência de 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
```

Saída:

```
{
  "JobId": "dg7289877365fc70299089371c043b96"
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StartSnomedctInferenceJob](#) na Referência de AWS CLI Comandos.

stop-entities-detection-v2-job

O código de exemplo a seguir mostra como usar `stop-entities-detection-v2-job`.

AWS CLI

Para interromper um trabalho de detecção de entidades

O `stop-entities-detection-v2-job` exemplo a seguir interrompe um trabalho assíncrono de detecção de entidades.

```
aws comprehendmedical stop-entities-detection-v2-job \
  --job-id "ab9887877365fe70299089371c043b96"
```

Saída:

```
{
  "JobId": "ab9887877365fe70299089371c043b96"
}
```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StopEntitiesDetectionV2Job na Referência](#) de AWS CLI Comandos.

stop-icd10-cm-inference-job

O código de exemplo a seguir mostra como usar `stop-icd10-cm-inference-job`.

AWS CLI

Para interromper um trabalho de ICD inferência de -10 CM

O `stop-icd10-cm-inference-job` exemplo a seguir interrompe um trabalho de ICD análise em lote de inferência de -10 CM.

```
aws comprehendmedical stop-icd10-cm-inference-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Saída:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96",  
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StopIcd10 CmInferenceJob](#) na Referência de AWS CLI Comandos.

stop-phi-detection-job

O código de exemplo a seguir mostra como usar `stop-phi-detection-job`.

AWS CLI

Para interromper um trabalho de detecção de informações de saúde protegidas (PHI)

O `stop-phi-detection-job` exemplo a seguir interrompe um trabalho assíncrono de detecção de informações de saúde protegidas (PHI).

```
aws comprehendmedical stop-phi-detection-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Saída:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Para obter mais informações, consulte [Batch APIs](#) no Guia do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StopPhiDetectionJob](#) na Referência de AWS CLI Comandos.

stop-rx-norm-inference-job

O código de exemplo a seguir mostra como usar `stop-rx-norm-inference-job`.

AWS CLI

Para interromper um trabalho de RxNorm inferência

O `stop-rx-norm-inference-job` exemplo a seguir interrompe um trabalho de ICD análise em lote de inferência de -10 CM.

```
aws comprehendmedical stop-rx-norm-inference-job \  
  --job-id "eg8199877365fc70299089371c043b96"
```

Saída:

```
{  
  "JobId": "eg8199877365fc70299089371c043b96",  
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StopRxNormInferenceJob](#) na Referência de AWS CLI Comandos.

stop-snomedct-inference-job

O código de exemplo a seguir mostra como usar `stop-snomedct-inference-job`.

AWS CLI

Para interromper um trabalho de inferência de SNOMED tomografia computadorizada

O `stop-snomedct-inference-job` exemplo a seguir interrompe um trabalho de análise em lote de inferência de SNOMED CT.

```
aws comprehendmedical stop-snomedct-inference-job \  
--job-id "8750034166436cdb52ffa3295a1b00a1"
```

Saída:

```
{  
  "JobId": "8750034166436cdb52ffa3295a1b00a1",  
}
```

Para obter mais informações, consulte [Ontologia que vincula a análise em lote no Guia](#) do desenvolvedor do Amazon Comprehend Medical.

- Para API obter detalhes, consulte [StopSnomedctInferenceJob](#) na Referência de AWS CLI Comandos.

AWS Config exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Config.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-config-rule

O código de exemplo a seguir mostra como usar `delete-config-rule`.

AWS CLI

Para excluir uma regra de AWS Config

O comando a seguir exclui uma regra de AWS Config chamada: MyConfigRule

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- Para API obter detalhes, consulte [DeleteConfigRule](#) na Referência de AWS CLI Comandos.

delete-delivery-channel

O código de exemplo a seguir mostra como usar delete-delivery-channel.

AWS CLI

Para excluir um canal de entrega

O comando a seguir exclui o canal de entrega padrão:

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- Para API obter detalhes, consulte [DeleteDeliveryChannel](#) na Referência de AWS CLI Comandos.

delete-evaluation-results

O código de exemplo a seguir mostra como usar delete-evaluation-results.

AWS CLI

Para excluir manualmente os resultados da avaliação

O comando a seguir exclui os resultados da avaliação atual da regra AWS gerenciada s3-bucket-versioning-enabled

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- Para API obter detalhes, consulte [DeleteEvaluationResults](#) na Referência de AWS CLI Comandos.

deliver-config-snapshot

O código de exemplo a seguir mostra como usar deliver-config-snapshot.

AWS CLI

Para fornecer um instantâneo de configuração

O comando a seguir fornece um snapshot de configuração para o bucket do Amazon S3 que pertence ao canal de entrega padrão:

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

Saída:

```
{
  "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- Para API obter detalhes, consulte [DeliverConfigSnapshot](#) na Referência de AWS CLI Comandos.

describe-compliance-by-config-rule

O código de exemplo a seguir mostra como usar `describe-compliance-by-config-rule`.

AWS CLI

Para obter informações de conformidade para suas regras de AWS Config

O comando a seguir retorna informações de conformidade para cada regra AWS do Config violada por um ou mais recursos: AWS

```
aws configservice describe-compliance-by-config-rule --compliance-  
types NON_COMPLIANT
```

Na saída, o valor de cada `CappedCount` atributo indica quantos recursos não estão em conformidade com a regra relacionada. Por exemplo, a saída a seguir indica que três recursos não estão em conformidade com a regra nomeada `InstanceTypesAreT2micro`.

Saída:

```
{
  "ComplianceByConfigRules": [
```

```

    {
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 3,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      },
      "ConfigRuleName": "InstanceTypesAreT2micro"
    },
    {
      "Compliance": {
        "ComplianceContributorCount": {
          "CappedCount": 10,
          "CapExceeded": false
        },
        "ComplianceType": "NON_COMPLIANT"
      },
      "ConfigRuleName": "RequiredTagsForVolumes"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeComplianceByConfigRule](#) na Referência de AWS CLI Comandos.

describe-compliance-by-resource

O código de exemplo a seguir mostra como usar `describe-compliance-by-resource`.

AWS CLI

Para obter informações de conformidade para seus AWS recursos

O comando a seguir retorna informações de conformidade de cada EC2 instância registrada pelo AWS Config e que viola uma ou mais regras:

```
aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
--compliance-types NON_COMPLIANT
```

Na saída, o valor de cada `CappedCount` atributo indica quantas regras o recurso viola. Por exemplo, a saída a seguir indica que a instância `i-1a2b3c4d` viola duas regras.

Saída:

```
{
  "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"
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeComplianceByResource](#) na Referência de AWS CLI Comandos.

describe-config-rule-evaluation-status

O código de exemplo a seguir mostra como usar `describe-config-rule-evaluation-status`.

AWS CLI

Para obter informações de status de uma regra AWS Config

O comando a seguir retorna as informações de status de uma regra AWS Config chamada: `MyConfigRule`

```
aws configservice describe-config-rule-evaluation-status --config-rule-  
names MyConfigRule
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeConfigRuleEvaluationStatus](#) na Referência de AWS CLI Comandos.

describe-config-rules

O código de exemplo a seguir mostra como usar `describe-config-rules`.

AWS CLI

Para obter detalhes de uma regra de AWS Config

O comando a seguir retorna detalhes de uma regra de AWS Config chamada:

`InstanceTypesAreT2micro`

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

Saída:

```
{  
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [DescribeConfigRules](#) na Referência de AWS CLI Comandos.

describe-configuration-recorder-status

O código de exemplo a seguir mostra como usar `describe-configuration-recorder-status`.

AWS CLI

Para obter informações de status do gravador de configuração

O comando a seguir retorna o status do gravador de configuração padrão:

```
aws configservice describe-configuration-recorder-status
```

Saída:

```
{
  "ConfigurationRecordersStatus": [
```

```
{
  "name": "default",
  "lastStatus": "SUCCESS",
  "recording": true,
  "lastStatusChangeTime": 1452193834.344,
  "lastStartTime": 1441039997.819,
  "lastStopTime": 1441039992.835
}
```

- Para API obter detalhes, consulte [DescribeConfigurationRecorderStatus](#) na Referência de AWS CLI Comandos.

describe-configuration-recorders

O código de exemplo a seguir mostra como usar `describe-configuration-recorders`.

AWS CLI

Para obter detalhes sobre o gravador de configuração

O comando a seguir retorna detalhes sobre o gravador de configuração padrão:

```
aws configservice describe-configuration-recorders
```

Saída:

```
{
  "ConfigurationRecorders": [
    {
      "recordingGroup": {
        "allSupported": true,
        "resourceTypes": [],
        "includeGlobalResourceTypes": true
      },
      "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-
A1B2C3D4E5F6",
      "name": "default"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeConfigurationRecorders](#) na Referência de AWS CLI Comandos.

describe-delivery-channel-status

O código de exemplo a seguir mostra como usar `describe-delivery-channel-status`.

AWS CLI

Para obter informações de status do canal de entrega

O comando a seguir retorna o status do canal de entrega:

```
aws configservice describe-delivery-channel-status
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeDeliveryChannelStatus](#) na Referência de AWS CLI Comandos.

describe-delivery-channels

O código de exemplo a seguir mostra como usar `describe-delivery-channels`.

AWS CLI

Para obter detalhes sobre o canal de entrega

O comando a seguir retorna detalhes sobre o canal de entrega:

```
aws configservice describe-delivery-channels
```

Saída:

```
{
  "DeliveryChannels": [
    {
      "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
      "name": "default",
      "s3BucketName": "config-bucket-123456789012"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeDeliveryChannels](#) na Referência de AWS CLI Comandos.

get-compliance-details-by-config-rule

O código de exemplo a seguir mostra como usar `get-compliance-details-by-config-rule`.

AWS CLI

Para obter os resultados da avaliação de uma regra AWS Config

O comando a seguir retorna os resultados da avaliação de todos os recursos que não estão em conformidade com uma regra de AWS Config chamada: `InstanceTypesAreT2micro`

```
aws configservice get-compliance-details-by-config-rule --config-rule-
name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [GetComplianceDetailsByConfigRule](#) na Referência de AWS CLI Comandos.

get-compliance-details-by-resource

O código de exemplo a seguir mostra como usar `get-compliance-details-by-resource`.

AWS CLI

Para obter os resultados da avaliação de um AWS recurso

O comando a seguir retorna os resultados da avaliação de cada regra com a qual a EC2 instância `i-1a2b3c4d` não está em conformidade:

```
aws configservice get-compliance-details-by-resource --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

Saída:

```
{
  "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"
  }
]
}
```

- Para API obter detalhes, consulte [GetComplianceDetailsByResource](#) na Referência de AWS CLI Comandos.

get-compliance-summary-by-config-rule

O código de exemplo a seguir mostra como usar `get-compliance-summary-by-config-rule`.

AWS CLI

Para obter o resumo de conformidade de suas regras de AWS Config

O comando a seguir retorna o número de regras que estão em conformidade e o número que não estão em conformidade:

```
aws configservice get-compliance-summary-by-config-rule
```

Na saída, o valor de cada `CappedCount` atributo indica quantas regras estão em conformidade ou não.

Saída:

```
{
  "ComplianceSummary": {
    "NonCompliantResourceCount": {
      "CappedCount": 3,
      "CapExceeded": false
    },
    "ComplianceSummaryTimestamp": 1452204131.493,
    "CompliantResourceCount": {
      "CappedCount": 2,
      "CapExceeded": false
    }
  }
}
```

```
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetComplianceSummaryByConfigRule](#) na Referência de AWS CLI Comandos.

get-compliance-summary-by-resource-type

O código de exemplo a seguir mostra como usar `get-compliance-summary-by-resource-type`.

AWS CLI

Para obter o resumo de conformidade de todos os tipos de recursos

O comando a seguir retorna o número de AWS recursos que não estão em conformidade e o número que estão em conformidade:

```
aws configservice get-compliance-summary-by-resource-type
```

Na saída, o valor de cada `CappedCount` atributo indica quantos recursos estão em conformidade ou não.

Saída:

```
{  
  "ComplianceSummariesByResourceType": [  
    {  
      "ComplianceSummary": {  
        "NonCompliantResourceCount": {  
          "CappedCount": 16,  
          "CapExceeded": false  
        },  
        "ComplianceSummaryTimestamp": 1453237464.543,  
        "CompliantResourceCount": {  
          "CappedCount": 10,  
          "CapExceeded": false  
        }  
      }  
    }  
  ]  
}
```

```
]
}
```

Para obter o resumo de conformidade de um tipo específico de recurso

O comando a seguir retorna o número de EC2 instâncias que não estão em conformidade e o número que estão em conformidade:

```
aws configservice get-compliance-summary-by-resource-type --resource-
types AWS::EC2::Instance
```

Na saída, o valor de cada CappedCount atributo indica quantos recursos estão em conformidade ou não.

Saída:

```
{
  "ComplianceSummariesByResourceType": [
    {
      "ResourceType": "AWS::EC2::Instance",
      "ComplianceSummary": {
        "NonCompliantResourceCount": {
          "CappedCount": 3,
          "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1452204923.518,
        "CompliantResourceCount": {
          "CappedCount": 7,
          "CapExceeded": false
        }
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [GetComplianceSummaryByResourceType](#) na Referência de AWS CLI Comandos.

get-resource-config-history

O código de exemplo a seguir mostra como usar get-resource-config-history.

AWS CLI

Para obter o histórico de configuração de um AWS recurso

O comando a seguir retorna uma lista de itens de configuração para uma EC2 instância com um ID de `i-1a2b3c4d`:

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d
```

- Para API obter detalhes, consulte [GetResourceConfigHistory](#) na Referência de AWS CLI Comandos.

get-status

O código de exemplo a seguir mostra como usar `get-status`.

AWS CLI

Para obter o status do AWS Config

O comando a seguir retorna o status do canal de entrega e do gravador de configuração:

```
aws configservice get-status
```

Saída:

```
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
```

- Para API obter detalhes, consulte [GetStatus](#) na Referência de AWS CLI Comandos.

list-discovered-resources

O código de exemplo a seguir mostra como usar `list-discovered-resources`.

AWS CLI

Para listar os recursos que o AWS Config descobriu

O comando a seguir lista as EC2 instâncias que o AWS Config descobriu:

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

Saída:

```
{
  "resourceIdentifiers": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-1a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-2a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-3a2b3c4d"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListDiscoveredResources](#) na Referência de AWS CLI Comandos.

put-config-rule

O código de exemplo a seguir mostra como usar `put-config-rule`.

AWS CLI

Para adicionar uma regra de Config AWS gerenciada

O comando a seguir fornece JSON código para adicionar uma regra Config AWS gerenciada:


```
aws configservice put-config-rule --config-rule file://RequiredTagsForEC2Instances.json
```

`RequiredTagsForEC2Instances.json` é um JSON arquivo que contém a configuração da regra:

```
{
  "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\"}"
}
```

Para o `ComplianceResourceTypes` atributo, esse JSON código limita o escopo a recursos do `AWS::EC2::Instance` tipo, então o AWS Config avaliará somente EC2 instâncias em relação à regra. Como a regra é uma regra gerenciada, o atributo `Owner` é definido como `AWS` e o atributo `SourceIdentifier` é definido como o identificador da regra, `REQUIRED_TAGS`. Para o atributo `InputParameters`, as chaves de tag exigidas pela regra, `CostCenter` e `Owner` são especificados.

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar a configuração da regra, execute o `describe-config-rules` comando e especifique o nome da regra.

Como adicionar uma regra do Config gerenciada pelo cliente

O comando a seguir fornece JSON código para adicionar uma regra Config gerenciada pelo cliente:

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

`InstanceTypesAreT2micro.json` é um JSON arquivo que contém a configuração da regra:

```
{
  "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\"}"
}
```

Para o `ComplianceResourceTypes` atributo, esse JSON código limita o escopo a recursos do `AWS::EC2::Instance` tipo, então o AWS Config avaliará somente EC2 instâncias em relação à regra. Como essa regra é uma regra gerenciada pelo cliente, o `Owner` atributo é definido como `CUSTOM_LAMBDA`, e o `SourceIdentifier` atributo é definido como, ARN da função AWS Lambda. O `SourceDetails` objeto é obrigatório. Os parâmetros especificados para o `InputParameters` atributo são passados para a função AWS Lambda quando o AWS Config a invoca para avaliar os recursos em relação à regra.

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar a configuração da regra, execute o `describe-config-rules` comando e especifique o nome da regra.

- Para API obter detalhes, consulte [PutConfigRule](#) na Referência de AWS CLI Comandos.

put-configuration-recorder

O código de exemplo a seguir mostra como usar `put-configuration-recorder`.

AWS CLI

Exemplo 1: Para registrar todos os recursos suportados

O comando a seguir cria um gravador de configuração que rastreia as alterações em todos os tipos de recursos compatíveis, incluindo os tipos de recursos globais:

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group allSupported=true,includeGlobalResourceTypes=true
```

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar as configurações do seu gravador de configuração, execute o `describe-configuration-recorders` comando.

Exemplo 2: Para registrar tipos específicos de recursos

O comando a seguir cria um gravador de configuração que rastreia as alterações somente nos tipos de recursos especificados no JSON arquivo para a opção `--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` é um JSON arquivo que especifica os tipos de recursos que o Config registrará: AWS

```
{  
  "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"  
  ]  
}
```

Antes de especificar os tipos de recursos para a `resourceTypes` chave, você deve definir as opções `allSupported` e `includeGlobalResourceTypes` como falsas ou omiti-las.

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar as configurações do seu gravador de configuração, execute o `describe-configuration-records` comando.

Exemplo 3: Para selecionar todos os recursos suportados, excluindo tipos específicos de recursos

O comando a seguir cria um gravador de configuração que rastreia as alterações em todos os tipos de recursos compatíveis atuais e futuros, excluindo os tipos de recursos especificados no JSON arquivo da opção `--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` é um JSON arquivo que especifica os tipos de recursos que o Config registrará: AWS

```
{  
  "allSupported": false,  
  "exclusionByResourceTypes": {  
    "resourceTypes": [  
      "AWS::Redshift::ClusterSnapshot",  
      "AWS::RDS::DBClusterSnapshot",  
      "AWS::CloudFront::StreamingDistribution"  
    ]  
  },  
  "includeGlobalResourceTypes": false,  
  "recordingStrategy": {  
    "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"  
  },  
}
```

Antes de especificar os tipos de recursos a serem excluídos da gravação: 1) Você deve definir as opções `allSupported` e `includeGlobalResourceTypes` como falsas ou omiti-las e 2) Você deve definir o `useOnly` campo de `RecordingStrategy` como `EXCLUSION_RESOURCE_BY_._TYPES`

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar as configurações do seu gravador de configuração, execute o `describe-configuration-records` comando.

- Para API obter detalhes, consulte [PutConfigurationRecorder](#) na Referência de AWS CLI Comandos.

put-delivery-channel

O código de exemplo a seguir mostra como usar `put-delivery-channel`.

AWS CLI

Para criar um canal de entrega

O comando a seguir fornece as configurações do canal de entrega como JSON código:

```
aws configservice put-delivery-channel --delivery-channel file://  
deliveryChannel.json
```

O `deliveryChannel.json` arquivo especifica os atributos do canal de entrega:

```
{  
  "name": "default",  
  "s3BucketName": "config-bucket-123456789012",  
  "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",  
  "configSnapshotDeliveryProperties": {  
    "deliveryFrequency": "Twelve_Hours"  
  }  
}
```

Esse exemplo define os seguintes atributos:

`name`- O nome do canal de entrega. Por padrão, o AWS Config atribui o nome `default` a um novo canal de entrega. Você não pode atualizar o nome do canal de entrega com o comando `put-delivery-channel`. Para ver as etapas para alterar o nome, consulte [Renomear o canal de entrega](#). `s3BucketName` - O nome do bucket do Amazon S3 para o qual o AWS Config entrega snapshots de configuração e arquivos de histórico de configuração. Se você especificar um bucket que pertença a outra AWS conta, esse bucket deve ter políticas que concedam permissões de acesso ao Config. AWS Para obter mais informações, consulte [Permissões para buckets do Amazon S3](#).

`snsTopicARN`- O Amazon Resource Name (ARN) do SNS tópico da Amazon para o qual o AWS Config envia notificações sobre alterações na configuração. Se você escolher um tópico de outra conta, o tópico deve ter políticas que concedam permissões de acesso ao Config. Para obter mais informações, consulte [Permissões para o SNS tópico da Amazon](#).

`configSnapshotDeliveryProperties`- Contém o `deliveryFrequency` atributo, que define com que frequência o AWS Config fornece instantâneos de configuração e com que frequência ele invoca avaliações para regras periódicas do Config.

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar as configurações do seu canal de entrega, execute o `describe-delivery-channels` comando.

- Para API obter detalhes, consulte [PutDeliveryChannel](#) na Referência de AWS CLI Comandos.

start-config-rules-evaluation

O código de exemplo a seguir mostra como usar `start-config-rules-evaluation`.

AWS CLI

Para executar uma avaliação sob demanda das regras do AWS Config

O comando a seguir inicia uma avaliação para duas regras AWS gerenciadas:

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-versioning-enabled cloudtrail-enabled
```

- Para API obter detalhes, consulte [StartConfigRulesEvaluation](#) na Referência de AWS CLI Comandos.

start-configuration-recorder

O código de exemplo a seguir mostra como usar `start-configuration-recorder`.

AWS CLI

Para iniciar o gravador de configuração

O comando a seguir inicia o gravador de configuração padrão:

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar se o AWS Config está gravando seus recursos, execute o comando `get-status`.

- Para API obter detalhes, consulte [StartConfigurationRecorder](#) na Referência de AWS CLI Comandos.

stop-configuration-recorder

O código de exemplo a seguir mostra como usar `stop-configuration-recorder`.

AWS CLI

Para parar o gravador de configuração

O comando a seguir interrompe o gravador de configuração padrão:

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

Se o comando for bem-sucedido, o AWS Config não retornará nenhuma saída. Para verificar se o AWS Config não está gravando seus recursos, execute o comando `get-status`.

- Para API obter detalhes, consulte [StopConfigurationRecorder](#) na Referência de AWS CLI Comandos.

subscribe

O código de exemplo a seguir mostra como usar `subscribe`.

AWS CLI

Para assinar o AWS Config

O comando a seguir cria o canal de entrega e o gravador de configuração padrão. O comando também especifica o bucket do Amazon S3 e o tópico da SNS Amazon para o qual o AWS Config fornecerá as informações de configuração:

```
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
```

Saída:

```
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"
  }
]
```

- Para API obter detalhes, consulte [Inscrever-se](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Connect usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Connect.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-user

O código de exemplo a seguir mostra como usar create-user.

AWS CLI

Para criar um usuário

O create-user exemplo a seguir adiciona um usuário com os atributos especificados à instância especificada do Amazon Connect.

```
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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Adicionar usuários](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

delete-user

O código de exemplo a seguir mostra como usar delete-user.

AWS CLI

Como excluir um usuário

O `delete-user` exemplo a seguir exclui o usuário especificado da instância especificada do Amazon Connect.

```
aws connect delete-user \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --user-id 87654321-2222-1234-1234-111234567891
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar usuários](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

describe-user-hierarchy-group

O código de exemplo a seguir mostra como usar `describe-user-hierarchy-group`.

AWS CLI

Para exibir os detalhes de um grupo hierárquico

O `describe-user-hierarchy-group` exemplo a seguir exibe os detalhes do grupo hierárquico especificado do Amazon Connect.

```
aws connect describe-user-hierarchy-group \  
  --hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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"
        }
    }
}

```

Para obter mais informações, consulte [Configurar hierarquias de agentes no Guia](#) do administrador do Amazon Connect.

- Para API obter detalhes, consulte [DescribeUserHierarchyGroup](#) na Referência de AWS CLI Comandos.

describe-user-hierarchy-structure

O código de exemplo a seguir mostra como usar `describe-user-hierarchy-structure`.

AWS CLI

Para exibir os detalhes de uma estrutura hierárquica

O `describe-user-hierarchy-structure` exemplo a seguir exibe os detalhes da estrutura hierárquica da instância especificada do Amazon Connect.

```

aws connect describe-user-hierarchy-group \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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"
    }
}
}
}

```

Para obter mais informações, consulte [Configurar hierarquias de agentes no Guia](#) do administrador do Amazon Connect.

- Para API obter detalhes, consulte [DescribeUserHierarchyStructure](#) na Referência de AWS CLI Comandos.

describe-user

O código de exemplo a seguir mostra como usar `describe-user`.

AWS CLI

Para exibir os detalhes de um usuário

O `describe-user` exemplo a seguir exibe os detalhes do usuário especificado do Amazon Connect.

```

aws connect describe-user \
  --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Saída:

```

{
  "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": {}
  }
}

```

Para obter mais informações, consulte [Gerenciar usuários](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [DescribeUser](#) na Referência de AWS CLI Comandos.

get-contact-attributes

O código de exemplo a seguir mostra como usar `get-contact-attributes`.

AWS CLI

Para recuperar os atributos de um contato

O `get-contact-attributes` exemplo a seguir recupera os atributos que foram definidos para o contato especificado do Amazon Connect.

```

aws connect get-contact-attributes \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7

```

Saída:

```
{
  "Attributes": {
    "greetingPlayed": "true"
  }
}
```

Para obter mais informações, consulte [Usar atributos de contato do Amazon Connect](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [GetContactAttributes](#) na Referência de AWS CLI Comandos.

list-contact-flows

O código de exemplo a seguir mostra como usar `list-contact-flows`.

AWS CLI

Para listar os fluxos de contato em uma instância

O `list-contact-flows` exemplo a seguir lista os fluxos de contato na instância especificada do Amazon Connect.

```
aws connect list-contact-flows \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
    },
]
}

```

Para obter mais informações, consulte [Criar fluxos de contato do Amazon Connect](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListContactFlows](#) na Referência de AWS CLI Comandos.

list-hours-of-operations

O código de exemplo a seguir mostra como usar `list-hours-of-operations`.

AWS CLI

Para listar as horas de operação em uma instância

O `list-hours-of-operations` exemplo a seguir lista as horas de operação da instância especificada do Amazon Connect.

```

aws connect list-hours-of-operations \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Saída:

```

{
  "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"
    }
  ]
}

```

```

    }
  ]
}

```

Para obter mais informações, consulte [Definir o horário de funcionamento de uma fila](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListHoursOfOperations](#) na Referência de AWS CLI Comandos.

list-phone-numbers

O código de exemplo a seguir mostra como usar `list-phone-numbers`.

AWS CLI

Para listar os números de telefone em uma instância

O `list-phone-numbers` exemplo a seguir lista os números de telefone na instância especificada do Amazon Connect.

```

aws connect list-phone-numbers \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Configurar números de telefone para sua central de contato](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListPhoneNumbers](#) na Referência de AWS CLI Comandos.

list-queues

O código de exemplo a seguir mostra como usar `list-queues`.

AWS CLI

Para listar as filas em uma instância

O `list-queues` exemplo a seguir lista as filas na instância especificada do Amazon Connect.

```

aws connect list-queues \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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"
},
]
}
```

Para obter mais informações, consulte [Criar uma fila](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListQueues](#) na Referência de AWS CLI Comandos.

list-routing-profiles

O código de exemplo a seguir mostra como usar `list-routing-profiles`.

AWS CLI

Para listar os perfis de roteamento em uma instância

O `list-routing-profiles` exemplo a seguir lista os perfis de roteamento na instância especificada do Amazon Connect.

```
aws connect list-routing-profiles \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
    },
]
}

```

Para obter mais informações, consulte [Criar um perfil de roteamento](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListRoutingProfiles](#) na Referência de AWS CLI Comandos.

list-security-profiles

O código de exemplo a seguir mostra como usar `list-security-profiles`.

AWS CLI

Para listar os perfis de segurança em uma instância

O `list-security-profiles` exemplo a seguir lista os perfis de segurança na instância especificada do Amazon Connect.

```

aws connect list-security-profiles \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Atribuir permissões: perfis de segurança](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListSecurityProfiles](#) na Referência de AWS CLI Comandos.

list-user-hierarchy-groups

O código de exemplo a seguir mostra como usar `list-user-hierarchy-groups`.

AWS CLI

Para listar os grupos de hierarquia de usuários em uma instância

O `list-user-hierarchy-groups` exemplo a seguir lista os grupos de hierarquia de usuários na instância especificada do Amazon Connect.

```

aws connect list-user-hierarchy-groups \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Saída:

```

{
  "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"
    },
]
}

```

Para obter mais informações, consulte [Configurar hierarquias de agentes no Guia](#) do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListUserHierarchyGroups](#) na Referência de AWS CLI Comandos.

list-users

O código de exemplo a seguir mostra como usar `list-users`.

AWS CLI

Para listar os grupos de hierarquia de usuários em uma instância

O `list-users` exemplo a seguir lista os usuários na instância especificada do Amazon Connect.

```

aws connect list-users \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Saída:

```

{
  "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"
}
]
```

Para obter mais informações, consulte [Adicionar usuários](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [ListUsers](#) na Referência de AWS CLI Comandos.

update-contact-attributes

O código de exemplo a seguir mostra como usar `update-contact-attributes`.

AWS CLI

Para atualizar o atributo de um contato

O `update-contact-attributes` exemplo a seguir atualiza o `greetingPlayed` atributo para o usuário especificado do Amazon Connect.

```
aws connect update-contact-attributes \
  --initial-contact-id 11111111-2222-3333-4444-12345678910 \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --attributes greetingPlayed=false
```

Este comando não produz saída.

Para obter mais informações, consulte [Usar atributos de contato do Amazon Connect](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [UpdateContactAttributes](#) na Referência de AWS CLI Comandos.

update-user-hierarchy

O código de exemplo a seguir mostra como usar `update-user-hierarchy`.

AWS CLI

Para atualizar a hierarquia de um usuário

O `update-user-hierarchy` exemplo a seguir atualiza a hierarquia do agente para o usuário especificado do Amazon Connect.

```
aws connect update-user-hierarchy \  
  --hierarchy-group-id 12345678-a1b2-c3d4-e5f6-123456789abc \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Este comando não produz saída.

Para obter mais informações, consulte [Definir as configurações do agente](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [UpdateUserHierarchy](#) na Referência de AWS CLI Comandos.

update-user-identity-info

O código de exemplo a seguir mostra como usar `update-user-identity-info`.

AWS CLI

Para atualizar as informações de identidade de um usuário

O `update-user-identity-info` exemplo a seguir atualiza as informações de identidade do usuário especificado do Amazon Connect.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Definir as configurações do agente](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [UpdateUserIdentityInfo](#) na Referência de AWS CLI Comandos.

update-user-phone-config

O código de exemplo a seguir mostra como usar `update-user-phone-config`.

AWS CLI

Para atualizar a configuração do telefone de um usuário

O `update-user-phone-config` exemplo a seguir atualiza a configuração do telefone para o usuário especificado.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Definir as configurações do agente](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [UpdateUserPhoneConfig](#) na Referência de AWS CLI Comandos.

update-user-routing-profile

O código de exemplo a seguir mostra como usar `update-user-routing-profile`.

AWS CLI

Para atualizar o perfil de roteamento de um usuário

O `update-user-routing-profile` exemplo a seguir atualiza o perfil de roteamento para o usuário especificado do Amazon Connect.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Definir as configurações do agente](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [UpdateUserRoutingProfile](#) na Referência de AWS CLI Comandos.

update-user-security-profiles

O código de exemplo a seguir mostra como usar update-user-security-profiles.

AWS CLI

Para atualizar os perfis de segurança de um usuário

O update-user-security-profiles exemplo a seguir atualiza o perfil de segurança do usuário especificado do Amazon Connect.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Atribuir permissões: perfis de segurança](#) no Guia do administrador do Amazon Connect.

- Para API obter detalhes, consulte [UpdateUserSecurityProfiles](#) na Referência de AWS CLI Comandos.

AWS Cost and Usage Report exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Cost and Usage Report.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-report-definition

O código de exemplo a seguir mostra como usar `delete-report-definition`.

AWS CLI

Para excluir um relatório de AWS custo e uso

Este exemplo exclui um relatório de AWS custo e uso.

Comando:

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- Para API obter detalhes, consulte [DeleteReportDefinition](#) na Referência de AWS CLI Comandos.

describe-report-definitions

O código de exemplo a seguir mostra como usar `describe-report-definitions`.

AWS CLI

Para recuperar uma lista de relatórios de AWS custo e uso

Este exemplo descreve uma lista de relatórios de AWS custo e uso pertencentes a uma conta.

Comando:

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

Saída:

```
{
  "ReportDefinitions": [
    {
      "ReportName": "ExampleReport",
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
        "REDSHIFT",
        "QUICKSIGHT"
      ],
      "AdditionalSchemaElements": [
        "RESOURCES"
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeReportDefinitions](#) na Referência de AWS CLI Comandos.

put-report-definition

O código de exemplo a seguir mostra como usar `put-report-definition`.

AWS CLI

Para criar relatórios de AWS custo e uso

O `put-report-definition` exemplo a seguir cria um relatório diário de AWS custo e uso que você pode carregar no Amazon Redshift ou na Amazon. QuickSight

```
aws cur put-report-definition --report-definition file://report-definition.json
```

Conteúdo de 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"
  ]
}
```

- Para API obter detalhes, consulte [PutReportDefinition](#) na Referência de AWS CLI Comandos.

Exemplos do Cost Explorer Service usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o serviço AWS Command Line Interface with Cost Explorer.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-cost-and-usage

O código de exemplo a seguir mostra como usar get-cost-and-usage.

AWS CLI

Para recuperar o uso do S3 de uma conta no mês de setembro de 2017

O `get-cost-and-usage` exemplo a seguir recupera o uso do S3 de uma conta no mês de setembro de 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
```

Conteúdo de `filters.json`:

```
{  
  "Dimensions": {  
    "Key": "SERVICE",  
    "Values": [  
      "Amazon Simple Storage Service"  
    ]  
  }  
}
```

Saída:

```
{  
  "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"
      }
    }
  }
}
```

```

    ]
  }
]
}

```

- Para API obter detalhes, consulte [GetCostAndUsage](#) na Referência de AWS CLI Comandos.

get-dimension-values

O código de exemplo a seguir mostra como usar `get-dimension-values`.

AWS CLI

Para recuperar as tags da dimensão `SERVICE`, com um valor de “Elastic”

Este exemplo recupera as tags da dimensão `SERVICE`, com um valor de “Elastic” de 01 de janeiro de 2017 a 18 de maio de 2017.

Comando:

```
aws ce get-dimension-values --search-string Elastic --time-period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

Saída:

```
{
  "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
}

```

- Para API obter detalhes, consulte [GetDimensionValues](#) na Referência de AWS CLI Comandos.

get-reservation-coverage

O código de exemplo a seguir mostra como usar `get-reservation-coverage`.

AWS CLI

Para recuperar a cobertura da reserva para instâncias EC2 t2.nano na região us-east-1

Este exemplo recupera a cobertura da reserva para instâncias EC2 t2.nano na região us-east-1 de julho a setembro de 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"
      ]
    }
  ]
}
```

Saída:

```
{
  "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
}
```

- Para API obter detalhes, consulte [GetReservationCoveragena](#) Referência de AWS CLI Comandos.

get-reservation-purchase-recommendation

O código de exemplo a seguir mostra como usar `get-reservation-purchase-recommendation`.

AWS CLI

Para recuperar as recomendações de reserva do Partial Upfront EC2 RIs com um prazo de três anos

O `get-reservation-purchase-recommendation` exemplo a seguir recupera recomendações para EC2 instâncias de pagamento adiantado parcial com um prazo de três anos, com base nos últimos 60 dias de uso. 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
```

Saída:

```
{
  "Recommendations": [],
  "Metadata": {
    "GenerationTimestamp": "2018-08-08T15:20:57Z",
    "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"
  }
}
```

- Para API obter detalhes, consulte [GetReservationPurchaseRecommendation](#)na Referência de AWS CLI Comandos.

get-reservation-utilization

O código de exemplo a seguir mostra como usar `get-reservation-utilization`.

AWS CLI

Para recuperar a utilização da reserva para sua conta

O `get-reservation-utilization` exemplo a seguir recupera a utilização da RI para todos os tipos de instância `t2.nano` de 01/03/2018 a 01/08/2018 da conta.

```
aws ce get-reservation-utilization \  
  --time-period Start=2018-03-01,End=2018-08-01 \  
  --filter file://filters.json
```

Conteúdo de `filters.json`:

```
{  
  "Dimensions": {  
    "Key": "INSTANCE_TYPE",  
    "Values": [  
      "t2.nano"  
    ]  
  }  
}
```

Saída:

```
{  
  "Total": {  
    "TotalAmortizedFee": "0",  
    "UtilizationPercentage": "0",  
    "PurchasedHours": "0",  
    "NetRISavings": "0",  
    "TotalActualHours": "0",  
    "AmortizedRecurringFee": "0",  
    "UnusedHours": "0",  
    "TotalPotentialRISavings": "0",  
    "OnDemandCostOfRIHoursUsed": "0",  
    "AmortizedUpfrontFee": "0"  
  },  
  "UtilizationsByTime": []  
}
```

- Para API obter detalhes, consulte [GetReservationUtilization](#) na Referência de AWS CLI Comandos.

get-tags

O código de exemplo a seguir mostra como usar `get-tags`.

AWS CLI

Para recuperar chaves e valores para uma tag de alocação de custos

Este exemplo recupera todas as tags de alocação de custos com uma chave de “Projeto” e um valor que contém “secretProject”.

Comando:

```
aws ce get-tags --search-string secretProject --time-period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

Saída:

```
{
  "ReturnSize": 2,
  "Tags": [
    "secretProject1",
    "secretProject2"
  ],
  "TotalSize": 2
}
```

- Para API obter detalhes, consulte [GetTags](#) na Referência de AWS CLI Comandos.

Exemplos de Firehose usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Firehose.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

list-delivery-streams

O código de exemplo a seguir mostra como usar `list-delivery-streams`.

AWS CLI

Para listar os fluxos de entrega disponíveis

O `list-delivery-streams` exemplo a seguir lista os fluxos de entrega disponíveis em sua AWS conta.

```
aws firehose list-delivery-streams
```

Saída:

```
{
  "DeliveryStreamNames": [
    "my-stream"
  ],
  "HasMoreDeliveryStreams": false
}
```

Para obter mais informações, consulte [Como criar um fluxo de entrega do Amazon Kinesis Firehose Data](#) no Guia do desenvolvedor do Amazon Kinesis Data Firehose.

- Para API obter detalhes, consulte [ListDeliveryStreams](#) na Referência de AWS CLI Comandos.

put-record-batch

O código de exemplo a seguir mostra como usar `put-record-batch`.

AWS CLI

Como gravar vários registros em um fluxo

O exemplo de `put-record-batch` a seguir grava três registros em um fluxo. Os dados são codificados no formato Base64.

```
aws firehose put-record-batch \
  --delivery-stream-name my-stream \
  --records file://records.json
```

Conteúdo de `myfile.json`:

```
[
  {"Data": "Rmlyc3QgdGhpbmc="},
  {"Data": "U2Vjb25kIHRoaW5n"},
  {"Data": "VGhpcmQgdGhpbmc="}
]
```

Saída:

```
{
  "FailedPutCount": 0,
  "Encrypted": false,
  "RequestResponses": [
    {
      "RecordId": "9D20J6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhY0aWKPSmZy/
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7Giixp0yLJkYZcae1i7HzlCEoy9LJhMr8EjDSi40m/9Vc2uhwwuAtGE0XKpxJ2WD7ZRwtAnY1KAnv
    },
    {
      "RecordId": "jFirejqxCLlK5xjH/UNmIMVcjkTEN76I7916X9PaZ
+PVa0SXDFu1WG0qEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GkmR5z9016XKJ
+ /+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP90iHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
    },
    {
      "RecordId":
"oy0amQ40o5Y2YV4vxzufdcM00w6n3EP13tpPJGoYVnKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3ie
DTBt3qBlmTj7Xq8SKVb01S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/s0kQLe+qh0sYKXWl"
    }
  ]
}
```

Para obter mais informações, consulte [Sending Data to an Amazon Kinesis Data Firehose Delivery Stream](#) no Guia do desenvolvedor do Amazon Kinesis Data Firehose.

- Para API obter detalhes, consulte [PutRecordBatch](#) na Referência de AWS CLI Comandos.

put-record

O código de exemplo a seguir mostra como usar `put-record`.

AWS CLI

Para gravar um registro em um stream

O `put-record` exemplo a seguir grava dados em um stream. Os dados são codificados no formato Base64.

```
aws firehose put-record \  
  --delivery-stream-name my-stream \  
  --record '{"Data": "SGVsbG8gd29ybGQ="}'
```

Saída:

```
{  
  "RecordId": "RjB5K/nnoGFHqwTsZ1Nd/  
TTqvjE8V5dsyXZTQn2JXrdpMT0wssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/  
j1Xujv31iDhnPdaLw4B0yM9Amv7PcCuB2079RuM0NhoakbyUymIwY8yt20G8X2420wu1j1Fafhci4erAt7QhDEvpwuK8  
  "Encrypted": false  
}
```

Para obter mais informações, consulte [Sending Data to an Amazon Kinesis Data Firehose Delivery Stream](#) no Guia do desenvolvedor do Amazon Kinesis Data Firehose.

- Para API obter detalhes, consulte [PutRecord](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Data Lifecycle Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Data Lifecycle Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-default-role

O código de exemplo a seguir mostra como usar `create-default-role`.

AWS CLI

Para criar a IAM função necessária para a Amazon DLM

O `d1m create-default-role` exemplo a seguir cria a função AWS `DataLifecycleManagerDefaultRole` padrão para gerenciar instantâneos.

```
aws d1m create-default-role \  
  --resource-type snapshot
```

Este comando não produz saída.

Para obter mais informações, consulte [Funções de serviço padrão para o Amazon Data Lifecycle Manager no](#) Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [CreateDefaultRole](#) na Referência de AWS CLI Comandos.

create-lifecycle-policy

O código de exemplo a seguir mostra como usar `create-lifecycle-policy`.

AWS CLI

Para criar uma política de ciclo de vida

O `create-lifecycle-policy` exemplo a seguir cria uma política de ciclo de vida que cria um instantâneo diário dos volumes no horário especificado. As tags especificadas são adicionadas aos instantâneos, e as tags também são copiadas do volume e adicionadas aos instantâneos. Se

a criação de um novo instantâneo exceder a contagem máxima especificada, o instantâneo mais antigo será excluído.

```
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
```

Conteúdo de `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  
      }  
    }  
  ]  
}
```

```
]
}
```

Saída:

```
{
  "PolicyId": "policy-0123456789abcdef0"
}
```

- Para API obter detalhes, consulte [CreateLifecyclePolicy](#) na Referência de AWS CLI Comandos.

delete-lifecycle-policy

O código de exemplo a seguir mostra como usar `delete-lifecycle-policy`.

AWS CLI

Para excluir uma política de ciclo de vida

O exemplo a seguir exclui a política de ciclo de vida especificada. :

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- Para API obter detalhes, consulte [DeleteLifecyclePolicy](#) na Referência de AWS CLI Comandos.

get-lifecycle-policies

O código de exemplo a seguir mostra como usar `get-lifecycle-policies`.

AWS CLI

Para obter um resumo de suas políticas de ciclo de vida

O `get-lifecycle-policies` exemplo a seguir lista todas as suas políticas de ciclo de vida.

```
aws dlm get-lifecycle-policies
```

Saída:

```
{
```

```

    "Policies": [
      {
        "PolicyId": "policy-0123456789abcdef0",
        "Description": "My first policy",
        "State": "ENABLED"
      }
    ]
  }

```

- Para API obter detalhes, consulte [GetLifecyclePolicies](#) na Referência de AWS CLI Comandos.

get-lifecycle-policy

O código de exemplo a seguir mostra como usar `get-lifecycle-policy`.

AWS CLI

Para descrever uma política de ciclo de vida

O `get-lifecycle-policy` exemplo a seguir exibe detalhes da política de ciclo de vida especificada.

```

aws dlm get-lifecycle-policy \
  --policy-id policy-0123456789abcdef0

```

Saída:

```

{
  "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
        }
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [GetLifecyclePolicy](#) na Referência de AWS CLI Comandos.

update-lifecycle-policy

O código de exemplo a seguir mostra como usar `update-lifecycle-policy`.

AWS CLI

Exemplo 1: Para habilitar uma política de ciclo de vida

O `update-lifecycle-policy` exemplo a seguir ativa a política de ciclo de vida especificada.

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --state ENABLED
```

Exemplo 2: Para desativar uma política de ciclo de vida

O `update-lifecycle-policy` exemplo a seguir desativa a política de ciclo de vida especificada.

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --state DISABLED
```

Exemplo 3: Para atualizar os detalhes da política de ciclo de vida

O `update-lifecycle-policy` exemplo a seguir atualiza as tags de destino para a política de ciclo de vida especificada.

```
aws dlm update-lifecycle-policy \  
  --policy-id policy-0123456789abcdef0 \  
  --policy-details file://policyDetails.json
```

Conteúdo de `policyDetails.json`. Outros detalhes não referenciados neste arquivo não são alterados pelo comando.

```
{  
  "TargetTags": [  
    {  
      "Key": "costCenter",  
      "Value": "120"  
    },  
    {  
      "Key": "project",  
      "Value": "lima"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [UpdateLifecyclePolicy](#) na Referência de AWS CLI Comandos.

AWS Data Pipeline exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Data Pipeline.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

activate-pipeline

O código de exemplo a seguir mostra como usar `activate-pipeline`.

AWS CLI

Para ativar um pipeline

Este exemplo ativa o pipeline especificado:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Para ativar o pipeline em uma data e hora específicas, use o seguinte comando:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --start-timestamp 2015-04-07T00:00:00Z
```

- Para API obter detalhes, consulte [ActivatePipeline](#) na Referência de AWS CLI Comandos.

add-tags

O código de exemplo a seguir mostra como usar `add-tags`.

AWS CLI

Para adicionar uma tag a um pipeline

Este exemplo adiciona a tag especificada ao pipeline especificado:

```
aws datapipeline add-tags --pipeline-id df-00627471S0VYZEXAMPLE --  
tags key=environment,value=production key=owner,value=sales
```

Para visualizar as tags, use o comando `describe-pipelines`. Por exemplo, as tags adicionadas no comando de exemplo aparecem da seguinte forma na saída de `describe-pipelines`:

```
{  
  ...  
  "tags": [  
    {  
      "value": "production",  
      "key": "environment"  
    },  
    {  
      "value": "sales",  
      "key": "owner"  
    }  
  ]  
  ...  
}
```

- Para API obter detalhes, consulte [AddTags](#) na Referência de AWS CLI Comandos.

create-pipeline

O código de exemplo a seguir mostra como usar `create-pipeline`.

AWS CLI

Para criar um pipeline

Este exemplo cria um pipeline:

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

A seguir está um exemplo de saída:

```
{
  "pipelineId": "df-00627471S0VYZEXAMPLE"
}
```

- Para API obter detalhes, consulte [CreatePipeline](#) na Referência de AWS CLI Comandos.

deactivate-pipeline

O código de exemplo a seguir mostra como usar deactivate-pipeline.

AWS CLI

Para desativar um pipeline

Este exemplo desativa o pipeline especificado:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Para desativar o pipeline somente após a conclusão de todas as atividades em execução, use o seguinte comando:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --no-cancel-active
```

- Para API obter detalhes, consulte [DeactivatePipeline](#) na Referência de AWS CLI Comandos.

delete-pipeline

O código de exemplo a seguir mostra como usar delete-pipeline.

AWS CLI

Para excluir um pipeline

Este exemplo exclui o pipeline especificado:

```
aws datapipeline delete-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

- Para API obter detalhes, consulte [DeletePipeline](#) na Referência de AWS CLI Comandos.

describe-pipelines

O código de exemplo a seguir mostra como usar `describe-pipelines`.

AWS CLI

Para descrever seus pipelines

Este exemplo descreve o pipeline especificado:

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471S0VYZEXAMPLE
```

A seguir está um exemplo de saída:

```
{
  "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": []
  }
]
}

```

- Para API obter detalhes, consulte [DescribePipelines](#) na Referência de AWS CLI Comandos.

get-pipeline-definition

O código de exemplo a seguir mostra como usar `get-pipeline-definition`.

AWS CLI

Para obter uma definição de pipeline

Este exemplo obtém a definição do pipeline para o pipeline especificado:

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE
```

A seguir está um exemplo de saída:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [GetPipelineDefinition](#) na Referência de AWS CLI Comandos.

list-pipelines

O código de exemplo a seguir mostra como usar `list-pipelines`.

AWS CLI

Para listar seus pipelines

Este exemplo lista seus pipelines:

```
aws datapipeline list-pipelines
```

A seguir está um exemplo de saída:

```
{
  "pipelineIdList": [
    {
      "id": "df-00627471S0VYZEXAMPLE",
      "name": "my-pipeline"
    },
    {
      "id": "df-09028963KNVMREXAMPLE",
      "name": "ImportDDB"
    },
    {
      "id": "df-0870198233ZYVEXAMPLE",
      "name": "CrossRegionDDB"
    },
    {
      "id": "df-00189603TB4MZEXAMPLE",
      "name": "CopyRedshift"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListPipelines](#) na Referência de AWS CLI Comandos.

list-runs

O código de exemplo a seguir mostra como usar `list-runs`.

AWS CLI

Exemplo 1: Para listar as execuções do seu pipeline

O `list-runs` exemplo a seguir lista as execuções do pipeline especificado.

```
aws datapipeline list-runs --pipeline-id df-00627471S0VYZEXAMPLE
```

Saída:

Name	Scheduled Start	Status	ID
	Started	Ended	
1. EC2ResourceObj	2015-04-12T17:33:02	CREATING	
@EC2ResourceObj_2015-04-12T17:33:02		2015-04-12T17:33:10	
2. S3InputLocation	2015-04-12T17:33:02	FINISHED	
@S3InputLocation_2015-04-12T17:33:02		2015-04-12T17:33:09	
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	

Exemplo 2: Para listar as execuções do pipeline entre as datas especificadas

O `list-runs` exemplo a seguir usa o `--start-interval` para especificar as datas a serem incluídas na saída.

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUK05 --start-interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- Para API obter detalhes, consulte [ListRuns](#) na Referência de AWS CLI Comandos.

put-pipeline-definition

O código de exemplo a seguir mostra como usar `put-pipeline-definition`.

AWS CLI

Para fazer o upload de uma definição de pipeline

Este exemplo carrega a definição do pipeline especificado para o pipeline especificado:

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --pipeline-definition file://my-pipeline-definition.json
```

A seguir está um exemplo de saída:

```
{
  "validationErrors": [],
  "errored": false,
```

```
"validationWarnings": []
}
```

- Para API obter detalhes, consulte [PutPipelineDefinition](#) na Referência de AWS CLI Comandos.

remove-tags

O código de exemplo a seguir mostra como usar `remove-tags`.

AWS CLI

Para remover uma tag de um pipeline

Esse exemplo remove a tag especificada do pipeline especificado:

```
aws datapipeline remove-tags --pipeline-id df-00627471S0VYZEXAMPLE --tag-  
keys environment
```

- Para API obter detalhes, consulte [RemoveTags](#) na Referência de AWS CLI Comandos.

DataSync exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with DataSync.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

update-location-azure-blob

O código de exemplo a seguir mostra como usar `update-location-azure-blob`.

AWS CLI

Para atualizar seu local de transferência com um novo agente

O `update-location-object-storage` exemplo a seguir atualiza sua DataSync localização para o Microsoft Azure Blob Storage com um novo 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" \  
  }'
```

Este comando não produz saída.

Para obter mais informações, consulte [Substituindo seu agente](#) no Guia AWS DataSync do usuário.

- Para API obter detalhes, consulte [UpdateLocationAzureBlob](#) na Referência de AWS CLI Comandos.

update-location-hdfs

O código de exemplo a seguir mostra como usar `update-location-hdfs`.

AWS CLI

Para atualizar seu local de transferência com um novo agente

O `update-location-hdfs` exemplo a seguir atualiza sua DataSync HDFS localização com um novo agente. Você só precisará das `--kerberos-krb5-conf` opções `--kerberos-keytab` e se seu HDFS cluster usar a autenticação 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-krb5-conf krb5.conf \  
  --kerberos-keytab keytab
```



```
--kerberos-keytab file://hdfs.keytab  
--kerberos-krb5-conf file://krb5.conf
```

Conteúdo de `hdfs.keytab`:

N/A. The content of this file is encrypted and not human readable.

Conteúdo de `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Substituindo seu agente](#) no Guia AWS DataSync do usuário.

- Para API obter detalhes, consulte [UpdateLocationHdfs](#) na Referência de AWS CLI Comandos.

update-location-nfs

O código de exemplo a seguir mostra como usar `update-location-nfs`.

AWS CLI

Para atualizar seu local de transferência com um novo agente

O `update-location-nfs` exemplo a seguir atualiza sua DataSync NFS localização com um novo 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Substituindo seu agente](#) no Guia AWS DataSync do usuário.

- Para API obter detalhes, consulte [UpdateLocationNfs](#) na Referência de AWS CLI Comandos.

update-location-object-storage

O código de exemplo a seguir mostra como usar `update-location-object-storage`.

AWS CLI

Para atualizar seu local de transferência com um novo agente

O `update-location-object-storage` exemplo a seguir atualiza seu local de armazenamento de DataSync objetos com um novo agente.

```
aws datasync update-location-object-storage \  
  --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
```

```
--agent-arns arn:aws:datasync:us-west-2:123456789012:agent/agent-1234567890abcdef0 \  
--secret-key secret-key-for-object-storage
```

Este comando não produz saída.

Para obter mais informações, consulte [Substituindo seu agente](#) no Guia AWS DataSync do usuário.

- Para API obter detalhes, consulte [UpdateLocationObjectStorage](#) na Referência de AWS CLI Comandos.

update-location-smb

O código de exemplo a seguir mostra como usar update-location-smb.

AWS CLI

Para atualizar seu local de transferência com um novo agente

O update-location-smb exemplo a seguir atualiza sua DataSync SMB localização com um novo agente.

```
aws datasync update-location-smb \  
--location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-abcdef01234567890 \  
--agent-arns arn:aws:datasync:us-west-2:123456789012:agent/agent-1234567890abcdef0 \  
--password smb-file-server-password
```

Este comando não produz saída.

Para obter mais informações, consulte [Substituindo seu agente](#) no Guia AWS DataSync do usuário.

- Para API obter detalhes, consulte [UpdateLocationSmb](#) na Referência de AWS CLI Comandos.

DAXexemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with DAX.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-cluster

O código de exemplo a seguir mostra como usar `create-cluster`.

AWS CLI

Para criar um DAX cluster

O `create-cluster` exemplo a seguir cria um DAX cluster com as configurações especificadas.

```
aws dax create-cluster \  
  --cluster-name daxcluster \  
  --node-type dax.r4.large \  
  --replication-factor 3 \  
  --iam-role-arn roleARN \  
  --sse-specification Enabled=true
```

Saída:

```
{  
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Etapa 3: Criar um DAX cluster](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [CreateCluster](#) na Referência de AWS CLI Comandos.

create-parameter-group

O código de exemplo a seguir mostra como usar `create-parameter-group`.

AWS CLI

Para criar um grupo de parâmetros

O exemplo `` `create-parameter-group` `` a seguir cria um grupo de parâmetros com as configurações especificadas.

```

aws dax create-parameter-group \
  --parameter-group-name daxparametergroup \
  --description "A new parameter group"

```

Saída:

```
{
  "ParameterGroup": {
    "ParameterGroupName": "daxparametergroup",
    "Description": "A new parameter group"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [CreateParameterGroup](#) na Referência de AWS CLI Comandos.

create-subnet-group

O código de exemplo a seguir mostra como usar `create-subnet-group`.

AWS CLI

Para criar um grupo de DAX sub-redes

O `create-subnet-group` exemplo a seguir cria um grupo de sub-redes com as configurações especificadas.

```
aws dax create-subnet-group \
  --subnet-group-name daxSubnetGroup \
  --subnet-ids subnet-11111111 subnet-22222222
```

Saída:

```
{
  "SubnetGroup": {
    "SubnetGroupName": "daxSubnetGroup",
    "VpcId": "vpc-05a1fa8e00c325226",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-11111111",
        "SubnetAvailabilityZone": "us-west-2b"
      },
      {
        "SubnetIdentifier": "subnet-22222222",
        "SubnetAvailabilityZone": "us-west-2c"
      }
    ]
  }
}
```

```

    }
  ]
}
}

```

Para obter mais informações, consulte [Etapa 2: Criar um grupo de sub-rede](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [CreateSubnetGroup](#) na Referência de AWS CLI Comandos.

decrease-replication-factor

O código de exemplo a seguir mostra como usar `decrease-replication-factor`.

AWS CLI

Para remover um ou mais nós do cluster

O `decrease-replication-factor` exemplo a seguir diminui o número de nós no DAX cluster especificado para um.

```

aws dax decrease-replication-factor \
  --cluster-name daxcluster \
  --new-replication-factor 1

```

Saída:

```

{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DecreaseReplicationFactor](#) na Referência de AWS CLI Comandos.

delete-cluster

O código de exemplo a seguir mostra como usar delete-cluster.

AWS CLI

Para excluir um DAX cluster

O delete-cluster exemplo a seguir exclui o DAX cluster especificado.

```
aws dax delete-cluster \
  --cluster-name daxcluster
```

Saída:

```
{
  "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"
    }
}
}

```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DeleteCluster](#) na Referência de AWS CLI Comandos.

delete-parameter-group

O código de exemplo a seguir mostra como usar delete-parameter-group.

AWS CLI

Para excluir um grupo de parâmetros

O delete-parameter-group exemplo a seguir exclui o grupo de DAX parâmetros especificado.

```
aws dax delete-parameter-group \
  --parameter-group-name daxparametergroup
```

Saída:

```
{
  "DeletionMessage": "Parameter group daxparametergroup has been deleted."
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DeleteParameterGroup](#) na Referência de AWS CLI Comandos.

delete-subnet-group

O código de exemplo a seguir mostra como usar `delete-subnet-group`.

AWS CLI

Para excluir um grupo de sub-redes

O `delete-subnet-group` exemplo a seguir exclui o grupo de DAX sub-rede especificado.

```
aws dax delete-subnet-group \
  --subnet-group-name daxSubnetGroup
```

Saída:

```
{
  "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DeleteSubnetGroup](#) na Referência de AWS CLI Comandos.

describe-clusters

O código de exemplo a seguir mostra como usar `describe-clusters`.

AWS CLI

Para retornar informações sobre todos os clusters provisionados DAX

O `describe-clusters` exemplo a seguir exibe detalhes sobre todos os clusters provisionados DAX

```
aws dax describe-clusters
```

Saída:

```
{
  "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"
    }
}
]
}

```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeClusters](#) na Referência de AWS CLI Comandos.

describe-default-parameters

O código de exemplo a seguir mostra como usar `describe-default-parameters`.

AWS CLI

Para retornar as informações padrão dos parâmetros do sistema para DAX

O `describe-default-parameters` exemplo a seguir exibe as informações padrão dos parâmetros do sistema para DAX.

```
aws dax describe-default-parameters
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeDefaultParameters](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para retornar todos os eventos relacionados a DAX clusters e grupos de parâmetros

O `describe-events` exemplo a seguir exibe detalhes de eventos relacionados a DAX clusters e grupos de parâmetros.

aws dax describe-events

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-parameter-groups

O código de exemplo a seguir mostra como usar `describe-parameter-groups`.

AWS CLI

Para descrever os grupos de parâmetros definidos em DAX

O `describe-parameter-groups` exemplo a seguir recupera detalhes sobre os grupos de parâmetros definidos em DAX.

```
aws dax describe-parameter-groups
```

Saída:

```
{
  "ParameterGroups": [
    {
      "ParameterGroupName": "default.dax1.0",
      "Description": "Default parameter group for dax1.0"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeParameterGroups](#) na Referência de AWS CLI Comandos.

describe-parameters

O código de exemplo a seguir mostra como usar `describe-parameters`.

AWS CLI

Para descrever os parâmetros definidos em um grupo de DAX parâmetros

O `describe-parameters` exemplo a seguir recupera detalhes sobre os parâmetros definidos no grupo de DAX parâmetros especificado.

```
aws dax describe-parameters \
```



```
--parameter-group-name default.dax1.0
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeParameters](#) na Referência de AWS CLI Comandos.

describe-subnet-groups

O código de exemplo a seguir mostra como usar `describe-subnet-groups`.

AWS CLI

Para descrever grupos de sub-redes definidos em DAX

O `describe-subnet-groups` exemplo a seguir recupera detalhes dos grupos de sub-redes definidos em DAX

```
aws dax describe-subnet-groups
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeSubnetGroups](#) na Referência de AWS CLI Comandos.

increase-replication-factor

O código de exemplo a seguir mostra como usar `increase-replication-factor`.

AWS CLI

Para aumentar o fator de replicação de um cluster DAX

O `increase-replication-factor` exemplo a seguir aumenta o fator de replicação do DAX cluster especificado para 3.

```
aws dax increase-replication-factor \  
  --cluster-name daxcluster \  
  --new-replication-factor 3
```

Saída:

```
{  
  "Cluster": {  
    "ClusterName": "daxcluster",  
    "ClusterArn": "arn:aws:dax:us-west-2:123456789012:cache/daxcluster",  
    "TotalNodes": 3,  
    "ActiveNodes": 1,  
    "NodeType": "dax.r4.large",  
    "Status": "modifying",  
    "ClusterDiscoveryEndpoint": {  
      "Address": "daxcluster.ey3o9d.clustercfg.dax.usw2.cache.amazonaws.com",  
      "Port": 8111  
    },  
    "Nodes": [  
      {  
        "NodeId": "daxcluster-a",  
        "Endpoint": {  
          "Address": "daxcluster-  
a.ey3o9d.0001.dax.usw2.cache.amazonaws.com",  
          "Port": 8111  
        },  
        "NodeCreateTime": 1576625059.509,  
        "AvailabilityZone": "us-west-2c",
```

```

        "NodeStatus": "available",
        "ParameterGroupStatus": "in-sync"
    },
    {
        "NodeId": "daxcluster-b",
        "NodeStatus": "creating"
    },
    {
        "NodeId": "daxcluster-c",
        "NodeStatus": "creating"
    }
],
"PreferredMaintenanceWindow": "thu:13:00-thu:14:00",
"SubnetGroup": "default",
"SecurityGroups": [
    {
        "SecurityGroupIdentifier": "sg-1af6e36e",
        "Status": "active"
    }
],
"IamRoleArn": "arn:aws:iam::123456789012:role/
DAXServiceRoleForDynamoDBAccess",
"ParameterGroup": {
    "ParameterGroupName": "default.dax1.0",
    "ParameterApplyStatus": "in-sync",
    "NodeIdsToReboot": []
},
"SSEDescription": {
    "Status": "ENABLED"
}
}
}

```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [IncreaseReplicationFactor](#) na Referência de AWS CLI Comandos.

list-tags

O código de exemplo a seguir mostra como usar `list-tags`.

AWS CLI

Para listar tags em um DAX recurso

O `list-tags` exemplo a seguir lista as chaves e os valores de tag anexados ao DAX cluster especificado.

```
aws dax list-tags \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "ClusterUsage",  
      "Value": "prod"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [ListTags](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um DAX recurso

O `tag-resource` exemplo a seguir anexa o nome da chave da tag especificada e o valor associado ao DAX cluster especificado para descrever o uso do cluster.

```
aws dax tag-resource \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \  
  --tags="Key=ClusterUsage,Value=prod"
```

Saída:

```
{
  "Tags": [
    {
      "Key": "ClusterUsage",
      "Value": "prod"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um DAX recurso

O `untag-resource` exemplo a seguir remove a tag com o nome de chave especificado de um DAX cluster.

```
aws dax untag-resource \
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
  --tag-keys="ClusterUsage"
```

Saída:

```
{
  "Tags": []
}
```

Para obter mais informações, consulte [Gerenciamento de DAX clusters](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

Exemplos de detetives usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Detective.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-invitation

O código de exemplo a seguir mostra como usar `accept-invitation`.

AWS CLI

Para aceitar um convite para se tornar uma conta membro em um gráfico de comportamento

O `accept-invitation` exemplo a seguir aceita um convite para se tornar uma conta membro no gráfico de comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`.

```
aws detective accept-invitation \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Este comando não produz saída.

Para obter mais informações, consulte [Respondendo a um convite de gráfico de comportamento](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [AcceptInvitation](#) na Referência de AWS CLI Comandos.

create-graph

O código de exemplo a seguir mostra como usar `create-graph`.

AWS CLI

Para habilitar o Amazon Detective e criar um novo gráfico de comportamento

O `create-graph` exemplo a seguir ativa o Detective para a AWS conta que executa o comando na região em que o comando é executado. É criado um novo gráfico de comportamento que tem essa conta como conta de administrador. O comando também atribui o valor Finance à tag Department.

```
aws detective create-graph \  
  --tags '{"Department": "Finance"}
```

Saída:

```
{  
  "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"  
}
```

Para obter mais informações, consulte [Habilitando o Amazon Detective](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [CreateGraph](#) na Referência de AWS CLI Comandos.

`create-members`

O código de exemplo a seguir mostra como usar `create-members`.

AWS CLI

Para convidar contas de membros para um gráfico de comportamento

O `create-members` exemplo a seguir convida duas AWS contas a se tornarem contas membros no gráfico de comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Para cada conta, a solicitação fornece o ID da AWS conta e o endereço de e-mail do usuário raiz da conta. A solicitação inclui uma mensagem personalizada para inserir no e-mail de convite.

```
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."
```

Saída:

```
{
  "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": [ ]
}
```

Para obter mais informações, consulte Convidar contas de membros para um gráfico de comportamento < <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html> > no Amazon Detective Administration Guide.

Para convidar contas de membros sem enviar e-mails de convite

O `create-members` exemplo a seguir convida duas AWS contas a se tornarem contas membros no gráfico de comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Para cada conta, a solicitação fornece o ID da AWS conta e o endereço de e-mail do usuário raiz da conta. As contas dos membros não recebem e-mails de convite.

```
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
```

Saída:

```
{
  "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": [ ]
}
```

Para obter mais informações, consulte Convidar contas de membros para um gráfico de comportamento < <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html> > no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [CreateMembers](#) na Referência de AWS CLI Comandos.

delete-graph

O código de exemplo a seguir mostra como usar `delete-graph`.

AWS CLI

Para desativar o Detective e excluir o gráfico de comportamento

O `delete-graph` exemplo a seguir desativa o Detective e exclui o gráfico de comportamento especificado.

```
aws detective delete-graph \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Este comando não produz saída.

Para obter mais informações, consulte [Desabilitar o Amazon Detective](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [DeleteGraph](#) na Referência de AWS CLI Comandos.

delete-members

O código de exemplo a seguir mostra como usar `delete-members`.

AWS CLI

Para remover contas de membros de um gráfico de comportamento

O `delete-members` exemplo a seguir remove duas contas membros do gráfico de comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Para identificar as contas, a solicitação fornece a AWS contaIDs.

```
aws detective delete-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Saída:

```
{  
  "AccountIds": [ "444455556666", "123456789012" ],
```

```
"UnprocessedAccounts": [ ]
}
```

Para obter mais informações, consulte [Removendo contas de membros de um gráfico de 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> > no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [DeleteMembers](#) na Referência de AWS CLI Comandos.

disassociate-membership

O código de exemplo a seguir mostra como usar `disassociate-membership`.

AWS CLI

Para renunciar à associação a partir de um gráfico de comportamento

O exemplo de `disassociate-membership` a seguir remove a AWS conta que executa o comando do gráfico de 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
```

Para obter mais informações, consulte [Removendo sua conta de um gráfico de 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> > no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [DisassociateMembership](#) na Referência de AWS CLI Comandos.

get-members

O código de exemplo a seguir mostra como usar `get-members`.

AWS CLI

Para recuperar informações sobre contas de membros do gráfico de comportamento selecionado

O `get-members` exemplo a seguir recupera informações sobre duas contas de membros no gráfico de comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Para as duas contas, a solicitação fornece a AWS `contaIDs`.

```
aws detective get-members \  
--account-ids 444455556666 123456789012 \  
--graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Saída:

```
{  
  "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": [ ]  
}
```

Para obter mais informações, consulte [Visualização da lista de contas em um gráfico de comportamento](https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html) < <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html> > no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [GetMembers](#) na Referência de AWS CLI Comandos.

list-graphs

O código de exemplo a seguir mostra como usar `list-graphs`.

AWS CLI

Para ver uma lista de gráficos de comportamento dos quais sua conta é administradora

O `list-graphs` exemplo a seguir recupera os gráficos de comportamento dos quais a conta de chamada é administradora na região atual.

```
aws detective list-graphs
```

Saída:

```
{
  "GraphList": [
    {
      "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "CreatedTime": 1579736111000
    }
  ]
}
```

- Para API obter detalhes, consulte [ListGraphs](#) na Referência de AWS CLI Comandos.

`list-invitations`

O código de exemplo a seguir mostra como usar `list-invitations`.

AWS CLI

Para ver uma lista de gráficos de comportamento dos quais uma conta é membro ou para a qual foi convidada

O `list-invitations` exemplo a seguir recupera os gráficos de comportamento para os quais a conta de chamada foi convidada. Os resultados incluem somente convites abertos e aceitos. Eles não incluem convites rejeitados ou assinaturas removidas.

```
aws detective list-invitations
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte Visualizar sua lista de convites para gráficos de comportamento < <https://docs.aws.amazon.com/detective/latest/adminguide/.html> > no Amazon Detective Administration Guidemember-view-graph-invitations.

- Para API obter detalhes, consulte [ListInvitations](#) na Referência de AWS CLI Comandos.

list-members

O código de exemplo a seguir mostra como usar `list-members`.

AWS CLI

Para listar as contas dos membros em um gráfico de comportamento

O `list-members` exemplo a seguir recupera as contas de membros convidadas e habilitadas para o gráfico `arn:aws:detective:us-east-1:111122223333:graph:123412341234` de comportamento. Os resultados não incluem contas de membros que foram removidas.

```
aws detective list-members \
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Saída:

```
{
  "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
  }
]
}

```

Para obter mais informações, consulte [Visualização da lista de contas em um gráfico de comportamento](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [ListMembers](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para recuperar as tags atribuídas a um gráfico de comportamento

O `list-tags-for-resource` exemplo a seguir retorna as tags atribuídas ao gráfico de comportamento especificado.

```

aws detective list-tags-for-resource \
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234

```


Saída:

```
{
  "Tags": {
    "Department" : "Finance"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de tags para um gráfico de comportamento](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

reject-invitation

O código de exemplo a seguir mostra como usar `reject-invitation`.

AWS CLI

Para rejeitar um convite para se tornar uma conta membro em um gráfico de comportamento

O `reject-invitation` exemplo a seguir rejeita um convite para se tornar uma conta membro no gráfico de comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`.

```
aws detective reject-invitation \
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Este comando não produz saída.

Para obter mais informações, consulte Respondendo a um convite gráfico de comportamento <<https://docs.aws.amazon.com/detective/latest/adminguide/> .html> no Amazon Detective Administration Guide `member-invitation-response`.

- Para API obter detalhes, consulte [RejectInvitation](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para atribuir uma tag a um recurso

O `tag-resource` exemplo a seguir atribui um valor para a tag `Department` ao gráfico de comportamento especificado.

```
aws detective tag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tags '{"Department":"Finance"}
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags para um gráfico de comportamento](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover um valor de tag de um recurso

O `untag-resource` exemplo a seguir remove a tag `Department` do gráfico de comportamento especificado.

```
aws detective untag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tag-keys "Department"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags para um gráfico de comportamento](#) no Amazon Detective Administration Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

Exemplos de Device Farm usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Device Farm.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-device-pool

O código de exemplo a seguir mostra como usar `create-device-pool`.

AWS CLI

Para criar um pool de dispositivos

O comando a seguir cria um pool de dispositivos Android para um projeto:

```
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"
```

Você pode obter o projeto a ARN partir da saída de `create-project` ou `list-projects`. O arquivo `device-pool-rules.json` é um JSON documento na pasta atual que especifica a plataforma do dispositivo:

```
[  
  {  
    "attribute": "PLATFORM",  
    "operator": "EQUALS",  
    "value": "\"ANDROID\""  
  }  
]
```

Saída:

```
{
```

```
"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"
}
```

- Para API obter detalhes, consulte [CreateDevicePool](#) na Referência de AWS CLI Comandos.

create-project

O código de exemplo a seguir mostra como usar `create-project`.

AWS CLI

Para criar um projeto

O comando a seguir cria um novo projeto chamado `my-project`:

```
aws devicefarm create-project --name my-project
```

Saída:

```
{
  "project": {
    "name": "myproject",
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
    "created": 1503612890.057
  }
}
```

- Para API obter detalhes, consulte [CreateProject](#) na Referência de AWS CLI Comandos.

create-upload

O código de exemplo a seguir mostra como usar `create-upload`.

AWS CLI

Para criar um upload

O comando a seguir cria um upload para um aplicativo 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
```

Você pode obter o projeto a ARN partir da saída de `create-project` ou `list-projects`.

Saída:

```
{
  "upload": {
    "status": "INITIALIZED",
    "name": "app.apk",
    "created": 1503614408.769,
    "url": "https://prod-us-west-2-uploads.s3-us-west-2.amazonaws.com/arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aproject%3A070fc3ca-c7e1-4471-91cf-d3e4efc50604/uploads/arn%3Aaws%3Adevicefarm%3Aus-west-2%3A123456789012%3Aupload%3A070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-ae9e-4087-09e6-f4cea3599514/app.apk?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20170824T224008Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs%32Faws4_request&X-Amz-Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f",
    "type": "ANDROID_APP",
    "arn": "arn:aws:devicefarm:us-west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-ae9e-4087-09e6-f4cea3599514"
  }
}
```

Use o sinal URL na saída para fazer upload de um arquivo para o 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=AKIAEXAMPLEPBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f"
```

- Para API obter detalhes, consulte [CreateUpload](#) na Referência de AWS CLI Comandos.

get-upload

O código de exemplo a seguir mostra como usar `get-upload`.

AWS CLI

Para ver um upload

O comando a seguir recupera informações sobre um upload:

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
```

Você pode obter o upload a ARN partir da saída de `create-upload`.

Saída:

```
{
  "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\\\"}"
  }
}
```

```
}
```

- Para API obter detalhes, consulte [GetUpload](#) na Referência de AWS CLI Comandos.

list-projects

O código de exemplo a seguir mostra como usar `list-projects`.

AWS CLI

Para listar projetos

O seguinte recupera uma lista de projetos:

```
aws devicefarm list-projects
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListProjects](#) na Referência de AWS CLI Comandos.

AWS Direct Connect exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Direct Connect.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-direct-connect-gateway-association-proposal

O código de exemplo a seguir mostra como usar `accept-direct-connect-gateway-association-proposal`.

AWS CLI

Para aceitar uma proposta de associação de gateway

O seguinte `accept-direct-connect-gateway-association-proposal` aceita a proposta especificada.

```
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"
}
]
```

Para obter mais informações, consulte [Aceitar ou rejeitar uma proposta da Transit Gateway Association](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [AcceptDirectConnectGatewayAssociationProposal](#) na Referência de AWS CLI Comandos.

allocate-connection-on-interconnect

O código de exemplo a seguir mostra como usar `allocate-connection-on-interconnect`.

AWS CLI

Para criar uma conexão hospedada em uma interconexão

O `allocate-connection-on-interconnect` comando a seguir cria uma conexão hospedada em uma interconexão:

```
aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101
```

Saída:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffzc51m1",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- Para API obter detalhes, consulte [AllocateConnectionOnInterconnect](#) na Referência de AWS CLI Comandos.

allocate-hosted-connection

O código de exemplo a seguir mostra como usar `allocate-hosted-connection`.

AWS CLI

Para criar uma conexão hospedada em uma interconexão

O `allocate-hosted-connection` exemplo a seguir cria uma conexão hospedada na interconexão especificada.

```
aws directconnect allocate-hosted-connection \  
  --bandwidth 500Mbps \  
  --connection-name mydcinterconnect \  
  --owner-account 123456789012 \  
  -connection-id dxcon-fgktov66 \  
  -vlan 101
```

Saída:

```
{  
  "partnerName": "TIVIT",  
  "vlan": 101,  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-ffzc51m1",  
  "connectionState": "ordering",  
  "bandwidth": "500Mbps",  
  "location": "TIVIT",  
  "connectionName": "mydcinterconnect",  
  "region": "sa-east-1"  
}
```

- Para API obter detalhes, consulte [AllocateHostedConnection](#) na Referência de AWS CLI Comandos.

allocate-private-virtual-interface

O código de exemplo a seguir mostra como usar `allocate-private-virtual-interface`.

AWS CLI

Para provisionar uma interface virtual privada

O `allocate-private-virtual-interface` comando a seguir provisiona uma interface virtual privada de propriedade de um cliente diferente:

```
aws directconnect allocate-private-virtual-interface --connection-id dxcon-ffjrkrx17 --owner-account 123456789012 --new-private-virtual-interface-allocation virtualInterfaceName=PrivateVirtualInterface,vlan=1000,asn=65000,authKey=asdf34ex
```

Saída:

```
{
  "virtualInterfaceState": "confirming",
  "asn": 65000,
  "vlan": 1000,
  "customerAddress": "192.168.1.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrkrx17",
  "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"
}
```

- Para API obter detalhes, consulte [AllocatePrivateVirtualInterface](#) na Referência de AWS CLI Comandos.

`allocate-public-virtual-interface`

O código de exemplo a seguir mostra como usar `allocate-public-virtual-interface`.

AWS CLI

Para provisionar uma interface virtual pública

O `allocate-public-virtual-interface` comando a seguir provisiona uma interface virtual pública de propriedade de um cliente diferente:

```
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}
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [AllocatePublicVirtualInterface](#) na Referência de AWS CLI Comandos.

allocate-transit-virtual-interface

O código de exemplo a seguir mostra como usar `allocate-transit-virtual-interface`.

AWS CLI

Para provisionar uma interface virtual de trânsito que pertença à AWS conta especificada

O `allocate-transit-virtual-interface` exemplo a seguir provisiona uma interface virtual de trânsito para a conta especificada.

```
aws directconnect allocate-transit-virtual-interface \
  --connection-id dxlag-fEXAMPLE \
  --owner-account 123456789012 \
  --new-transit-virtual-interface-allocation "virtualInterfaceName=Example Transit
Virtual
Interface, vlan=126, asn=65110, mtu=1500, authKey=0xzxgA9YoW9h58u8SEXAMPLE, amazonAddress=192.16"
```

Saída:

```
{
  "virtualInterface": {
    "ownerAccount": "123456789012",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "Example Transit Virtual Interface",
    "vlan": 126,
    "asn": 65110,
    "amazonSideAsn": 7224,
    "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
    "amazonAddress": "192.168.1.1/30",
    "customerAddress": "192.168.1.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "confirming",
    "customerRouterConfig": "<?xml version='1.0' encoding=
'UTF-8'>\n<logical_connection id='dxvif-fEXAMPLE'>\n  <vlan>126</
vlan>\n  <customer_address>192.168.1.2/30</customer_address>\n
```

```

<amazon_address>192.168.1.1/30</amazon_address>\n  <bgp_asn>65110</bgp_asn>\n
<bgp_auth_key>0xzxcgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
  "mtu": 1500,
  "jumboFrameCapable": true,
  "virtualGatewayId": "",
  "directConnectGatewayId": "",
  "routeFilterPrefixes": [],
  "bgpPeers": [
    {
      "bgpPeerId": "dxpeer-fEXAMPLE",
      "asn": 65110,
      "authKey": "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"
    }
  ]
}
}

```

Para obter mais informações, consulte [Criação de uma interface virtual de trânsito hospedado](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [AllocateTransitVirtualInterface](#) na Referência de AWS CLI Comandos.

associate-connection-with-lag

O código de exemplo a seguir mostra como usar `associate-connection-with-lag`.

AWS CLI

Para associar uma conexão a um LAG

O exemplo a seguir associa a conexão especificada à especificadaLAG.

Comando:

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --  
connection-id dxcon-fg9607vm
```

Saída:

```
{  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-fg9607vm",  
  "lagId": "dxlag-fhccu14t",  
  "connectionState": "requested",  
  "bandwidth": "1Gbps",  
  "location": "EqDC2",  
  "connectionName": "Con2ForLag",  
  "region": "us-east-1"  
}
```

- Para API obter detalhes, consulte [AssociateConnectionWithLag](#) na Referência de AWS CLI Comandos.

associate-hosted-connection

O código de exemplo a seguir mostra como usar `associate-hosted-connection`.

AWS CLI

Para associar uma conexão hospedada a um LAG

O exemplo a seguir associa a conexão hospedada especificada à especificadaLAG.

Comando:

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t  
--connection-id dxcon-fg9607vm
```

Saída:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- Para API obter detalhes, consulte [AssociateHostedConnection](#) na Referência de AWS CLI Comandos.

associate-virtual-interface

O código de exemplo a seguir mostra como usar `associate-virtual-interface`.

AWS CLI

Para associar uma interface virtual a uma conexão

O exemplo a seguir associa a interface virtual especificada à especificada LAG. Como alternativa, para associar a interface virtual a uma conexão, especifique o ID de uma conexão AWS Direct Connect para `--connection-id`; por exemplo, `dxcon-ffnikghc`.

Comando:

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --
virtual-interface-id dxvif-fgputw0j
```

Saída:

```
{
  "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"
}

```

- Para API obter detalhes, consulte [AssociateVirtualInterface](#) na Referência de AWS CLI Comandos.

confirm-connection

O código de exemplo a seguir mostra como usar `confirm-connection`.

AWS CLI

Para confirmar a criação de uma conexão hospedada em uma interconexão

O `confirm-connection` comando a seguir confirma a criação de uma conexão hospedada em uma interconexão:

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

Saída:

```
{
  "connectionState": "pending"
}
```

- Para API obter detalhes, consulte [ConfirmConnection](#) na Referência de AWS CLI Comandos.

confirm-private-virtual-interface

O código de exemplo a seguir mostra como usar `confirm-private-virtual-interface`.

AWS CLI

Para aceitar a propriedade de uma interface virtual privada

O `confirm-private-virtual-interface` comando a seguir aceita a propriedade de uma interface virtual privada criada por outro cliente:

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

Saída:

```
{
  "virtualInterfaceState": "pending"
}
```

- Para API obter detalhes, consulte [ConfirmPrivateVirtualInterface](#) na Referência de AWS CLI Comandos.

confirm-public-virtual-interface

O código de exemplo a seguir mostra como usar `confirm-public-virtual-interface`.

AWS CLI

Para aceitar a propriedade de uma interface virtual pública

O `confirm-public-virtual-interface` comando a seguir aceita a propriedade de uma interface virtual pública criada por outro cliente:

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

Saída:

```
{
  "virtualInterfaceState": "verifying"
}
```

- Para API obter detalhes, consulte [ConfirmPublicVirtualInterface](#) na Referência de AWS CLI Comandos.

confirm-transit-virtual-interface

O código de exemplo a seguir mostra como usar `confirm-transit-virtual-interface`.

AWS CLI

Para aceitar a propriedade de uma interface virtual de trânsito

O seguinte `confirm-transit-virtual-interface` aceita a propriedade de uma interface virtual de trânsito criada por outro cliente.

```
aws directconnect confirm-transit-virtual-interface \
  --virtual-interface-id dxvif-fEXAMPLE \
```

```
--direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```

Saída:

```
{  
  "virtualInterfaceState": "pending"  
}
```

Para obter mais informações, consulte [Aceitando uma interface virtual hospedada](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [ConfirmTransitVirtualInterface](#) na Referência de AWS CLI Comandos.

create-bgp-peer

O código de exemplo a seguir mostra como usar create-bgp-peer.

AWS CLI

Para criar uma sessão de IPv6 BGP emparelhamento

O exemplo a seguir cria uma sessão de IPv6 BGP emparelhamento em uma interface dxvif-fg1vuj3d virtual privada. Os IPv6 endereços dos pares são alocados automaticamente pela Amazon.

Comando:

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

Saída:

```
{  
  "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_ukbCer16EYA0uexample",
    "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_ukbCer16EYA0uexample</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"
}
}

```

- Para API obter detalhes, consulte [CreateBgpPeerna](#) Referência de AWS CLI Comandos.

create-connection

O código de exemplo a seguir mostra como usar `create-connection`.

AWS CLI

Para criar uma conexão da sua rede com um local do AWS Direct Connect

O `create-connection` comando a seguir cria uma conexão da sua rede com um local do AWS Direct Connect:

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-name "Connection to AWS"
```

Saída:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "connectionName": "Connection to AWS",
  "region": "sa-east-1"
}
```

- Para API obter detalhes, consulte [CreateConnection](#) na Referência de AWS CLI Comandos.

create-direct-connect-gateway-association-proposal

O código de exemplo a seguir mostra como usar `create-direct-connect-gateway-association-proposal`.

AWS CLI

Para criar uma proposta para associar o gateway de trânsito especificado ao gateway Direct Connect especificado

O `create-direct-connect-gateway-association-proposal` exemplo a seguir cria uma proposta que associa o gateway de trânsito especificado ao gateway Direct Connect especificado.

```
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
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Criação de uma proposta da Transit Gateway Association](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [CreateDirectConnectGatewayAssociationProposal](#) na Referência de AWS CLI Comandos.

create-direct-connect-gateway-association

O código de exemplo a seguir mostra como usar `create-direct-connect-gateway-association`.

AWS CLI

Para associar um gateway privado virtual a um gateway Direct Connect

O exemplo a seguir associa o gateway privado virtual `vgw-6efe725e` ao gateway Direct Connect. `5f294f92-bafb-4011-916d-9b0bexample` Você deve executar o comando na região em que o gateway privado virtual está localizado.

Comando:

```
aws directconnect create-direct-connect-gateway-association --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

Saída:

```
{
  "directConnectGatewayAssociation": {
    "associationState": "associating",
    "virtualGatewayOwnerAccount": "123456789012",
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
    "virtualGatewayId": "vgw-6efe725e",
    "virtualGatewayRegion": "us-east-2"
  }
}
```

- Para API obter detalhes, consulte [CreateDirectConnectGatewayAssociation](#) na Referência de AWS CLI Comandos.

create-direct-connect-gateway

O código de exemplo a seguir mostra como usar `create-direct-connect-gateway`.

AWS CLI

Para criar um gateway Direct Connect

O exemplo a seguir cria um gateway Direct Connect com o nome `DxGateway1`.

Comando:

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-name "DxGateway1"
```

Saída:


```
{
  "directConnectGateway": {
    "amazonSideAsn": 64512,
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
    "ownerAccount": "123456789012",
    "directConnectGatewayName": "DxGateway1",
    "directConnectGatewayState": "available"
  }
}
```

- Para API obter detalhes, consulte [CreateDirectConnectGateway](#) na Referência de AWS CLI Comandos.

create-interconnect

O código de exemplo a seguir mostra como usar `create-interconnect`.

AWS CLI

Para criar uma interconexão entre a rede de um parceiro e AWS

O `create-interconnect` comando a seguir cria uma interconexão entre a rede de um parceiro do AWS Direct Connect e um local específico do AWS Direct Connect:

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
--bandwidth 1Gbps --location TIVIT
```

Saída:

```
{
  "region": "sa-east-1",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "interconnectName": "1G Interconnect to AWS",
  "interconnectId": "dxcon-fgktov66",
  "interconnectState": "requested"
}
```

- Para API obter detalhes, consulte [CreateInterconnect](#) na Referência de AWS CLI Comandos.

create-lag

O código de exemplo a seguir mostra como usar `create-lag`.

AWS CLI

Para criar um LAG com novas conexões

O exemplo a seguir cria LAG e solicita duas novas conexões AWS Direct Connect para o LAG com uma largura de banda de 1 Gbps.

Comando:

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --  
connections-bandwidth 1Gbps --lag-name 1GBLag
```

Saída:

```
{  
  "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-ffjhj9lx",
"minimumLinks": 0,
"connectionsBandwidth": "1Gbps",
"region": "us-east-1",
"location": "CSVA1"
}

```

Para criar um LAG usando uma conexão existente

O exemplo a seguir cria um LAG a partir de uma conexão existente em sua conta e solicita uma segunda nova conexão para a LAG com a mesma largura de banda e localização da conexão existente.

Comando:

```

aws directconnect create-lag --location EqDC2 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dx

```

Saída:

```

{
  "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"
}

```

- Para API obter detalhes, consulte [CreateLag](#) na Referência de AWS CLI Comandos.

create-private-virtual-interface

O código de exemplo a seguir mostra como usar `create-private-virtual-interface`.

AWS CLI

Para criar uma interface virtual privada

O `create-private-virtual-interface` comando a seguir cria uma interface virtual privada:

```

aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrnx17 --
new-private-virtual-
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam
aba37db6

```

Saída:

```

{
  "virtualInterfaceState": "pending",
  "asn": 65000,
  "vlan": 101,
  "customerAddress": "192.168.1.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrnx17",

```

```

    "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"
  }

```

- Para API obter detalhes, consulte [CreatePrivateVirtualInterface](#) na Referência de AWS CLI Comandos.

create-public-virtual-interface

O código de exemplo a seguir mostra como usar create-public-virtual-interface.

AWS CLI

Para criar uma interface virtual pública

O create-public-virtual-interface comando a seguir cria uma interface virtual pública:

```

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}

```

Saída:

```

{
  "virtualInterfaceState": "verifying",
  "asn": 65000,
  "vlan": 2000,
  "customerAddress": "203.0.113.2/30",
  "ownerAccount": "123456789012",

```

```

"connectionId": "dxcon-ffjrkrx17",
"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"
}

```

- Para API obter detalhes, consulte [CreatePublicVirtualInterface](#) na Referência de AWS CLI Comandos.

create-transit-virtual-interface

O código de exemplo a seguir mostra como usar create-transit-virtual-interface.

AWS CLI

Para criar uma interface virtual de trânsito

O create-transit-virtual-interface exemplo a seguir cria uma interface virtual de trânsito para a conexão especificada.

```

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}]"

```

Saída:

```
{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Criação de uma interface virtual de trânsito para o Direct Connect Gateway](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [CreateTransitVirtualInterface](#) na Referência de AWS CLI Comandos.

delete-bgp-peer

O código de exemplo a seguir mostra como usar `delete-bgp-peer`.

AWS CLI

Para excluir um BGP par de uma interface virtual

O exemplo a seguir exclui o IPv6 BGP peer da interface virtual. `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

```

Saída:

```

{
  "virtualInterface": {
    "virtualInterfaceState": "available",
    "asn": 65000,
    "vlan": 125,
    "customerAddress": "169.254.255.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fguhmq1c",
    "addressFamily": "ipv4",
  }
}

```



```

"virtualGatewayId": "vgw-f9eb0c90",
"virtualInterfaceId": "dxvif-fg1vuj3d",
"authKey": "0xC_ukbCerl6EYA0example",
"routeFilterPrefixes": [],
"location": "EqDC2",
"bgpPeers": [
  {
    "bgpStatus": "down",
    "customerAddress": "169.254.255.2/30",
    "addressFamily": "ipv4",
    "authKey": "0xC_ukbCerl6EYA0uexample",
    "bgpPeerState": "available",
    "amazonAddress": "169.254.255.1/30",
    "asn": 65000
  },
  {
    "bgpStatus": "down",
    "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
    "addressFamily": "ipv6",
    "authKey": "0xS27kAIU_VHPjjAexample",
    "bgpPeerState": "deleting",
    "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
    "asn": 64600
  }
],
"customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
  <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
  <bgp_auth_key>0xC_ukbCerl6EYA0example</bgp_auth_key>\n  <amazon_bgp_asn>7224</
amazon_bgp_asn>\n  <connection_type>private</connection_type>\n</logical_connection>
\n",
  "amazonAddress": "169.254.255.1/30",
  "virtualInterfaceType": "private",
  "virtualInterfaceName": "Test"
}
}

```

- Para API obter detalhes, consulte [DeleteBgpPeerna](#) Referência de AWS CLI Comandos.

delete-connection

O código de exemplo a seguir mostra como usar delete-connection.

AWS CLI

Para excluir uma conexão

O `delete-connection` comando a seguir exclui a conexão especificada:

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

Saída:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "deleted",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "connectionName": "Connection to AWS",
  "region": "sa-east-1"
}
```

- Para API obter detalhes, consulte [DeleteConnection](#) na Referência de AWS CLI Comandos.

`delete-direct-connect-gateway-association`

O código de exemplo a seguir mostra como usar `delete-direct-connect-gateway-association`.

AWS CLI

Para excluir uma associação de gateway Direct Connect

O `delete-direct-connect-gateway-association` exemplo a seguir exclui a associação do gateway Direct Connect com um gateway de trânsito que tem a ID de associação especificada.

```
aws directconnect delete-direct-connect-gateway-association --association-id be85116d-46eb-4b43-a27a-da0c2ad648de
```

Saída:

```
{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Associando e desassociando gateways de transporte público](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [DeleteDirectConnectGatewayAssociation](#) na Referência de AWS CLI Comandos.

delete-direct-connect-gateway

O código de exemplo a seguir mostra como usar `delete-direct-connect-gateway`.

AWS CLI

Para excluir um gateway Direct Connect

O exemplo a seguir exclui o gateway `5f294f92-bafb-4011-916d-9b0bexample` Direct Connect.

Comando:

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Saída:

```
{
```

```
"directConnectGateway": {
  "amazonSideAsn": 64512,
  "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
  "ownerAccount": "123456789012",
  "directConnectGatewayName": "DxGateway1",
  "directConnectGatewayState": "deleting"
}
}
```

- Para API obter detalhes, consulte [DeleteDirectConnectGateway](#) na Referência de AWS CLI Comandos.

delete-interconnect

O código de exemplo a seguir mostra como usar delete-interconnect.

AWS CLI

Para excluir uma interconexão

O delete-interconnect comando a seguir exclui a interconexão especificada:

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

Saída:

```
{
  "interconnectState": "deleted"
}
```

- Para API obter detalhes, consulte [DeleteInterconnect](#) na Referência de AWS CLI Comandos.

delete-lag

O código de exemplo a seguir mostra como usar delete-lag.

AWS CLI

Para excluir um LAG

O exemplo a seguir exclui o especificadoLAG.

Comando:

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [DeleteLag](#) na Referência de AWS CLI Comandos.

delete-virtual-interface

O código de exemplo a seguir mostra como usar delete-virtual-interface.

AWS CLI

Para excluir uma interface virtual

O delete-virtual-interface comando a seguir exclui a interface virtual especificada:

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhkh74f
```

Saída:

```
{
  "virtualInterfaceState": "deleting"
}
```

- Para API obter detalhes, consulte [DeleteVirtualInterface](#) na Referência de AWS CLI Comandos.

describe-connection-loa

O código de exemplo a seguir mostra como usar `describe-connection-loa`.

AWS CLI

Para descrever seu LOA - CFA para uma conexão usando Linux ou Mac OS X

O exemplo a seguir descreve sua conexão LOA - CFA `fordxcon-fh6ayh1d`. O conteúdo do LOA - CFA é codificado em base64. Esse comando usa os `--query` parâmetros `--output` e para controlar a saída e extrair o conteúdo da `loaContent` estrutura. A parte final do comando decodifica o conteúdo usando o base64 utilitário e envia a saída para um PDF arquivo.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --  
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Para descrever sua LOA - CFA para uma conexão usando o Windows

O exemplo anterior requer o uso do base64 utilitário para decodificar a saída. Em um computador Windows, você pode usar `certutil` em vez disso. No exemplo a seguir, o primeiro comando descreve sua conexão LOA - CFA `fordxcon-fh6ayh1d` e usa os `--query` parâmetros `--output` e para controlar a saída e extrair o conteúdo da `loaContent` estrutura em um arquivo chamado `myLoaCfa.base64`. O segundo comando usa o `certutil` utilitário para decodificar o arquivo e enviar a saída para um PDF arquivo.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --  
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Para obter mais informações sobre como controlar a AWS CLI saída, consulte [Controlando a saída de AWS comando da interface de linha](#) de comando no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DescribeConnectionLoa](#) na Referência de AWS CLI Comandos.

describe-connections-on-interconnect

O código de exemplo a seguir mostra como usar `describe-connections-on-interconnect`.

AWS CLI

Para listar conexões em uma interconexão

O `describe-connections-on-interconnect` comando a seguir lista as conexões que foram provisionadas na interconexão especificada:

```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-fgktov66
```

Saída:

```
{
  "connections": [
    {
      "partnerName": "TIVIT",
      "vlan": 101,
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffzc51m1",
      "connectionState": "ordering",
      "bandwidth": "500Mbps",
      "location": "TIVIT",
      "connectionName": "mydcinterconnect",
      "region": "sa-east-1"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeConnectionsOnInterconnect](#) na Referência de AWS CLI Comandos.

`describe-connections`

O código de exemplo a seguir mostra como usar `describe-connections`.

AWS CLI

Para listar todas as conexões na região atual

O `describe-connections` comando a seguir lista todas as conexões na região atual:

```
aws directconnect describe-connections
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeConnections](#) na Referência de AWS CLI Comandos.

describe-direct-connect-gateway-association-proposals

O código de exemplo a seguir mostra como usar `describe-direct-connect-gateway-association-proposals`.

AWS CLI

Para descrever suas propostas de associação de gateway Direct Connect

O `describe-direct-connect-gateway-association-proposals` exemplo a seguir exibe detalhes sobre suas propostas de associação de gateway Direct Connect.

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

Saída:

```
{
  "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"
    }
  ]
}
```

```

    }
  ]
}

```

Para obter mais informações, consulte [Associando e desassociando gateways de transporte público](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [DescribeDirectConnectGatewayAssociationProposals](#) na Referência de AWS CLI Comandos.

describe-direct-connect-gateway-associations

O código de exemplo a seguir mostra como usar `describe-direct-connect-gateway-associations`.

AWS CLI

Para descrever as associações de gateway Direct Connect

O exemplo a seguir descreve todas as associações com o 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
```

Saída:

```
{
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoiOU830TFodzycnZCbkn4MExHeHVwQT09IiwieYyI6InIxTEN0UEVHV0I1UF1kaWFnN1",
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeDirectConnectGatewayAssociations](#) na Referência de AWS CLI Comandos.

describe-direct-connect-gateway-attachments

O código de exemplo a seguir mostra como usar `describe-direct-connect-gateway-attachments`.

AWS CLI

Para descrever os anexos do gateway Direct Connect

O exemplo a seguir descreve as interfaces virtuais conectadas ao gateway Direct Connect `5f294f92-bafb-4011-916d-9b0bexample`.

Comando:

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Saída:

```
{
  "directConnectGatewayAttachments": [
    {
      "virtualInterfaceOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualInterfaceRegion": "us-east-2",
      "attachmentState": "attaching",
      "virtualInterfaceId": "dxvif-fg9zyabc"
    }
  ]
}
```

```
    }
  ],
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoibEhXd1NpUXF5RzhoL1JyUW52S1V2QT09IiwiaYyI6Im5wQjFHQ0RyQUdRS3puNnNXcU
}
```

- Para API obter detalhes, consulte [DescribeDirectConnectGatewayAttachments](#) na Referência de AWS CLI Comandos.

describe-direct-connect-gateways

O código de exemplo a seguir mostra como usar `describe-direct-connect-gateways`.

AWS CLI

Para descrever seus gateways Direct Connect

O exemplo a seguir descreve todos os seus gateways Direct Connect.

Comando:

```
aws directconnect describe-direct-connect-gateways
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [DescribeDirectConnectGateways](#) na Referência de AWS CLI Comandos.

describe-hosted-connections

O código de exemplo a seguir mostra como usar `describe-hosted-connections`.

AWS CLI

Para listar conexões em uma interconexão

O exemplo a seguir lista as conexões que foram provisionadas em uma determinada interconexão.

Comando:

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

Saída:

```
{
  "connections": [
    {
      "partnerName": "TIVIT",
      "vlan": 101,
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffzc51m1",
      "connectionState": "ordering",
      "bandwidth": "500Mbps",
      "location": "TIVIT",
      "connectionName": "mydcinterconnect",
      "region": "sa-east-1"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeHostedConnections](#) na Referência de AWS CLI Comandos.

describe-interconnect-loa

O código de exemplo a seguir mostra como usar `describe-interconnect-loa`.

AWS CLI

Para descrever seu LOA - CFA para uma interconexão usando Linux ou Mac OS X

O exemplo a seguir descreve sua LOA - CFA para interconexão `dxcon-fh6ayh1d`. O conteúdo do LOA - CFA é codificado em base64. Esse comando usa os `--query` parâmetros `--output` e para controlar a saída e extrair o conteúdo da `loaContent` estrutura. A parte final do comando decodifica o conteúdo usando o base64 utilitário e envia a saída para um PDF arquivo.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --  
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Para descrever seu LOA - CFA para uma interconexão usando o Windows

O exemplo anterior requer o uso do base64 utilitário para decodificar a saída. Em um computador Windows, você pode usar `certutil` em vez disso. No exemplo a seguir, o primeiro comando descreve sua interconexão LOA - CFA for `dxcon-fh6ayh1d` e usa os `--query` parâmetros `--output` e para controlar a saída e extrair o conteúdo da `loaContent` estrutura em um arquivo chamado `myLoaCfa.base64`. O segundo comando usa o `certutil` utilitário para decodificar o arquivo e enviar a saída para um PDF arquivo.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --  
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Para obter mais informações sobre como controlar a AWS CLI saída, consulte [Controlando a saída de AWS comando da interface de linha](#) de comando no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DescribeInterconnectLoa](#) na Referência de AWS CLI Comandos.

describe-interconnects

O código de exemplo a seguir mostra como usar `describe-interconnects`.

AWS CLI

Para listar as interconexões

O `describe-interconnects` comando a seguir lista as interconexões pertencentes à sua AWS conta:

```
aws directconnect describe-interconnects
```

Saída:

```
{
  "interconnects": [
    {
      "region": "sa-east-1",
      "bandwidth": "1Gbps",
      "location": "TIVIT",
      "interconnectName": "1G Interconnect to AWS",
      "interconnectId": "dxcon-fgktov66",
      "interconnectState": "down"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeInterconnects](#) na Referência de AWS CLI Comandos.

describe-lags

O código de exemplo a seguir mostra como usar `describe-lags`.

AWS CLI

Para descrever seu LAGs

O comando a seguir descreve todos os seus LAGs para a região atual.

Comando:

```
aws directconnect describe-lags
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLags](#) na Referência de AWS CLI Comandos.

describe-loa

O código de exemplo a seguir mostra como usar `describe-loa`.

AWS CLI

Para descrever seu LOA - CFA para uma conexão usando Linux ou Mac OS X

O exemplo a seguir descreve sua conexão LOA - CFA `fordxcon-fh6ayh1d`. O conteúdo do LOA - CFA é codificado em base64. Esse comando usa os `--query` parâmetros `--output` e `--query` para controlar a saída e extrair o conteúdo da `loaContent` estrutura. A parte final do comando decodifica o conteúdo usando o `base64` utilitário e envia a saída para um PDF arquivo.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query loa.loaContent/base64 --decode > myLoaCfa.pdf
```

Para descrever sua LOA - CFA para uma conexão usando o Windows

O exemplo anterior requer o uso do `base64` utilitário para decodificar a saída. Em um computador Windows, você pode usar `certutil` em vez disso. No exemplo a seguir, o primeiro comando descreve sua conexão LOA - CFA `for dxcon-fh6ayh1d` e usa os `--query` parâmetros `--output` e `--query` para controlar a saída e extrair o conteúdo da `loaContent` estrutura em um arquivo chamado `myLoaCfa.base64`. O segundo comando usa o `certutil` utilitário para decodificar o arquivo e enviar a saída para um PDF arquivo.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Para obter mais informações sobre como controlar a AWS CLI saída, consulte [Controlando a saída de AWS comando da interface de linha](#) de comando no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DescribeLoa](#) na Referência de AWS CLI Comandos.

describe-locations

O código de exemplo a seguir mostra como usar `describe-locations`.

AWS CLI

Para listar parceiros e locais do AWS Direct Connect

O `describe-locations` comando a seguir lista os parceiros e locais do AWS Direct Connect na região atual:

```
aws directconnect describe-locations
```

Saída:

```
{
  "locations": [
    {
      "locationName": "NAP do Brasil, Barueri, Sao Paulo",
      "locationCode": "TNDB"
    },
    {
      "locationName": "Tivit - Site Transamerica (Sao Paulo)",
      "locationCode": "TIVIT"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLocations](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever tags para seus recursos do AWS Direct Connect

O comando a seguir descreve as tags da conexão `dxcon-abcabc12`.

Comando:

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12
```

Saída:

```
{
  "resourceTags": [
    {
      "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-
abcabc12",
      "tags": [
        {
          "value": "VAConnection",
          "key": "Name"
        }
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

describe-virtual-gateways

O código de exemplo a seguir mostra como usar `describe-virtual-gateways`.

AWS CLI

Para listar gateways privados virtuais

O `describe-virtual-gateways` comando a seguir lista os gateways privados virtuais pertencentes à sua AWS conta:

```
aws directconnect describe-virtual-gateways
```

Saída:

```
{
  "virtualGateways": [
    {
      "virtualGatewayId": "vgw-aba37db6",
      "virtualGatewayState": "available"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeVirtualGateways](#) na Referência de AWS CLI Comandos.

describe-virtual-interfaces

O código de exemplo a seguir mostra como usar `describe-virtual-interfaces`.

AWS CLI

Para listar todas as interfaces virtuais

O `describe-virtual-interfaces` comando a seguir lista as informações sobre todas as interfaces virtuais associadas à sua AWS conta:

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrnx17
```

Saída:

```
{
  "virtualInterfaces": [
    {
      "virtualInterfaceState": "down",
      "asn": 65000,
      "vlan": 101,
      "customerAddress": "192.168.1.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffjrnx17",
      "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeVirtualInterfaces](#) na Referência de AWS CLI Comandos.

disassociate-connection-from-lag

O código de exemplo a seguir mostra como usar `disassociate-connection-from-lag`.

AWS CLI

Para desassociar uma conexão de um LAG

O exemplo a seguir dissocia a conexão especificada da especificada. LAG

Comando:

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --  
connection-id dxcon-fg9607vm
```

Saída:

```
{  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-fg9607vm",  
  "connectionState": "requested",  
  "bandwidth": "1Gbps",  
  "location": "EqDC2",  
  "connectionName": "Con2ForLag",  
  "region": "us-east-1"  
}
```

- Para API obter detalhes, consulte [DisassociateConnectionFromLag](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para adicionar uma tag a um recurso do AWS Direct Connect

O comando a seguir adiciona uma tag com uma chave de Name e um valor de VAConnection à conexão dxcon-abcabc12. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-  
east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso do AWS Direct Connect

O comando a seguir remove a tag com a chave `Name` da conexão `dxcon-abcabc12`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-direct-connect-gateway-association

O código de exemplo a seguir mostra como usar `update-direct-connect-gateway-association`.

AWS CLI

Para atualizar os atributos especificados da associação do gateway Direct Connect

O `update-direct-connect-gateway-association` exemplo a seguir adiciona o CIDR bloco especificado a uma associação de 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
```

Saída:

```
{  
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com gateways Direct Connect](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [UpdateDirectConnectGatewayAssociation](#) na Referência de AWS CLI Comandos.

update-lag

O código de exemplo a seguir mostra como usar `update-lag`.

AWS CLI

Para atualizar um LAG

O exemplo a seguir altera o nome do especificado LAG.

Comando:

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

Saída:

```
{
  "awsDevice": "CSVA1-23u8t1paz8iks",
  "numberOfConnections": 2,
  "lagState": "down",
```



```

"ownerAccount": "123456789012",
"lagName": "2ConnLag",
"connections": [
  {
    "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"
  },
  {
    "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"
  }
],
"lagId": "dxlag-ffjhj91x",
"minimumLinks": 0,
"connectionsBandwidth": "1Gbps",
"region": "us-east-1",
"location": "CSVA1"
}

```

- Para API obter detalhes, consulte [UpdateLag](#) na Referência de AWS CLI Comandos.

update-virtual-interface-attributes

O código de exemplo a seguir mostra como usar `update-virtual-interface-attributes`.

AWS CLI

Para atualizar o MTU de uma interface virtual

O `update-virtual-interface-attributes` exemplo a seguir MTU atualiza a interface virtual especificada.

```
aws directconnect update-virtual-interface-attributes \
  --virtual-interface-id dxvif-fEXAMPLE \
  --mtu 1500
```

Saída:

```
{
  "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  \n  <customer_address>169.254.248.2/30</customer_address>\n  \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": []
  }
```

Para obter mais informações, consulte [Configurando a rede MTU para interfaces virtuais privadas ou interfaces virtuais de trânsito](#) no Guia do usuário do AWS Direct Connect.

- Para API obter detalhes, consulte [UpdateVirtualInterfaceAttributes](#) na Referência de AWS CLI Comandos.

AWS Directory Service exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Directory Service.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-directories

O código de exemplo a seguir mostra como usar `describe-directories`.

AWS CLI

Para obter detalhes sobre seus diretórios

O `describe-directories` exemplo a seguir exibe detalhes sobre o diretório especificado.

```
aws ds describe-directories \  
  --directory-id d-a1b2c3d4e5
```

Saída:

```
{  
  "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"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeDirectories](#) na Referência de AWS CLI Comandos.

describe-trusts

O código de exemplo a seguir mostra como usar `describe-trusts`.

AWS CLI

Para obter detalhes sobre suas relações de confiança

O `describe-trusts` exemplo a seguir exibe detalhes sobre as relações de confiança do diretório especificado.

```
aws ds describe-trusts \
  --directory-id d-a1b2c3d4e5
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeTrusts](#) na Referência de AWS CLI Comandos.

AWS DMS exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS DMS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-resource

O código de exemplo a seguir mostra como usar `add-tags-to-resource`.

AWS CLI

Para adicionar tags a um recurso

O `add-tags-to-resource` exemplo a seguir adiciona tags a uma instância de replicação.

```
aws dms add-tags-to-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  \  
  --tags Key=Environment,Value=PROD Key=Project,Value=dbMigration
```

Este comando não produz saída.

Para obter mais informações, consulte [Tagging Resources](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [AddTagsToResource](#) na Referência de AWS CLI Comandos.

create-endpoint

O código de exemplo a seguir mostra como usar `create-endpoint`.

AWS CLI

Para criar um endpoint

O `create-endpoint` exemplo a seguir cria um endpoint para uma fonte do Amazon S3.

```
aws dms create-endpoint \  
  --endpoint-type source \  
  --engine-name s3 \  
  --endpoint-identifier src-endpoint \  
  --s3-settings file://s3-settings.json
```

Conteúdo de `s3-settings.json`:

```
{  
  "BucketName": "my-corp-data",  
  "BucketFolder": "sourcedata",  
  "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role"  
}
```

Saída:

```
{  
  "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
    }
}
}

```

Para obter mais informações, consulte [Trabalhando com AWS DMS endpoints](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [CreateEndpoint](#) na Referência de AWS CLI Comandos.

create-event-subscription

O código de exemplo a seguir mostra como usar `create-event-subscription`.

AWS CLI

Para listar assinaturas de eventos

O `create-event-subscription` exemplo a seguir cria uma assinatura de evento para um SNS tópico da 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

```

Saída:

```

{
  "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
  }
}

```


Para obter mais informações, consulte [Trabalhando com eventos e notificações](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [CreateEventSubscription](#) na Referência de AWS CLI Comandos.

create-replication-instance

O código de exemplo a seguir mostra como usar `create-replication-instance`.

AWS CLI

Para criar uma instância de replicação

O `create-replication-instance` exemplo a seguir cria uma instância de replicação.

```
aws dms create-replication-instance \  
  --replication-instance-identifier my-repl-instance \  
  --replication-instance-class dms.t2.micro \  
  --allocated-storage 5
```

Saída:

```
{  
  "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
  }
}

```

Para obter mais informações, consulte Como [trabalhar com uma instância de AWS DMS replicação](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [CreateReplicationInstance](#) na Referência de AWS CLI Comandos.

create-replication-subnet-group

O código de exemplo a seguir mostra como usar `create-replication-subnet-group`.

AWS CLI

Para criar um grupo de sub-redes

O `create-replication-subnet-group` exemplo a seguir cria um grupo que consiste em 3 sub-redes.

```

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

```

Saída:

```

{
  "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"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Configurando uma rede para uma instância de replicação](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [CreateReplicationSubnetGroup](#) na Referência de AWS CLI Comandos.

create-replication-task

O código de exemplo a seguir mostra como usar `create-replication-task`.

AWS CLI

Para criar uma tarefa de replicação

O `create-replication-task` exemplo a seguir cria uma tarefa de replicação.

```
aws dms create-replication-task \
```

```

--replication-task-identifier moveit2 \
--source-endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
--target-endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U \
--replication-instance-arn $RI_ARN \
--migration-type full-load \
--table-mappings file://table-mappings.json

```

Conteúdo de `table-mappings.json`:

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "prodrep",
        "table-name": "%"
      },
      "rule-action": "include",
      "filters": []
    }
  ]
}

```

Saída:

```

{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": "...output omitted...",
    "ReplicationTaskSettings": "...output omitted...",
    "Status": "creating",
    "ReplicationTaskCreationDate": 1590524772.505,
  }
}

```

```

    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Para obter mais informações, consulte [Trabalhando com AWS DMS tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [CreateReplicationTask](#) na Referência de AWS CLI Comandos.

delete-connection

O código de exemplo a seguir mostra como usar `delete-connection`.

AWS CLI

Para excluir uma conexão

O `delete-connection` exemplo a seguir dissocia um endpoint de uma instância de replicação.

```

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

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte https://docs.aws.amazon.com/dms/latest/userguide/CHAP__endpoints.creating.html no Guia do Usuário do Database Migration Service.AWS

- Para API obter detalhes, consulte [DeleteConnection](#) na Referência de AWS CLI Comandos.

delete-endpoint

O código de exemplo a seguir mostra como usar delete-endpoint.

AWS CLI

Para excluir um endpoint

O delete-endpoint exemplo a seguir exclui um endpoint.

```
aws dms delete-endpoint \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:0UJJVX04XZ4CYTSEG5XGMN2R3Y
```

Saída:

```
{  
  "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
```

```
    }  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com AWS DMS endpoints](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DeleteEndpoint](#) na Referência de AWS CLI Comandos.

delete-event-subscription

O código de exemplo a seguir mostra como usar `delete-event-subscription`.

AWS CLI

Para excluir uma assinatura de evento

O `delete-event-subscription` exemplo a seguir exclui uma assinatura de um SNS tópico da Amazon.

```
aws dms delete-event-subscription \  
  --subscription-name "my-dms-events"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com eventos e notificações](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DeleteEventSubscription](#) na Referência de AWS CLI Comandos.

delete-replication-instance

O código de exemplo a seguir mostra como usar `delete-replication-instance`.

AWS CLI

Para excluir uma instância de replicação

O exemplo da `delete-replication-instance` a seguir exclui uma instância de replicação.

```
aws dms delete-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte Como [trabalhar com uma instância de AWS DMS replicação](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DeleteReplicationInstance](#) na Referência de AWS CLI Comandos.

delete-replication-subnet-group

O código de exemplo a seguir mostra como usar delete-replication-subnet-group.

AWS CLI

Para excluir um grupo de sub-redes

O delete-replication-subnet-group exemplo a seguir exclui um grupo de sub-redes.

```
aws dms delete-replication-subnet-group \
--replication-subnet-group-identifier my-subnet-group
```

Saída:

```
(none)
```

Para obter mais informações, consulte [Configurando uma rede para uma instância de replicação](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DeleteReplicationSubnetGroup](#) na Referência de AWS CLI Comandos.

delete-replication-task

O código de exemplo a seguir mostra como usar delete-replication-task.

AWS CLI

Para excluir uma tarefa de replicação

O delete-replication-task exemplo a seguir exclui uma tarefa de replicação.

```
aws dms delete-replication-task \  
  --replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Saída:

```
{  
  "ReplicationTask": {  
    "ReplicationTaskIdentifier": "moveit2",  
    "SourceEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",  
    "TargetEndpointArn": "arn:aws:dms:us-  
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",  
    "ReplicationInstanceArn": "arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "MigrationType": "full-load",  
    "TableMappings": "...output omitted...",  
    "ReplicationTaskSettings": "...output omitted...",  
    "Status": "deleting",  
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",  
    "ReplicationTaskCreationDate": 1590524772.505,  
    "ReplicationTaskStartDate": 1590789988.677,  
    "ReplicationTaskArn": "arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com AWS DMS tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DeleteReplicationTask](#) na Referência de AWS CLI Comandos.

describe-account-attributes

O código de exemplo a seguir mostra como usar `describe-account-attributes`.

AWS CLI

Como descrever os atributos da conta

O `describe-account-attributes` exemplo a seguir lista os atributos da sua AWS conta.

```
aws dms describe-account-attributes
```

Saída:

```
{
  "AccountQuotas": [
    {
      "AccountQuotaName": "ReplicationInstances",
      "Used": 1,
      "Max": 20
    },
    {
      "AccountQuotaName": "AllocatedStorage",
      "Used": 5,
      "Max": 10000
    },
    ...remaining output omitted...
  ],
  "UniqueAccountIdentifier": "cqahfbfy5xee"
}
```

- Para API obter detalhes, consulte [DescribeAccountAttributes](#) na Referência de AWS CLI Comandos.

describe-certificates

O código de exemplo a seguir mostra como usar `describe-certificates`.

AWS CLI

Para listar os certificados disponíveis

O `describe-certificates` exemplo a seguir lista os certificados disponíveis em sua AWS conta.

```
aws dms describe-certificates
```

Saída:

```
{
  "Certificates": [
    {
      "CertificateIdentifier": "my-cert",
      "CertificateCreationDate": 1543259542.506,
      "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"
      ... remaining output omitted ...
    }
  ]
}
```

Para obter mais informações, consulte [Usando SSL](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeCertificates](#) na Referência de AWS CLI Comandos.

describe-connections

O código de exemplo a seguir mostra como usar `describe-connections`.

AWS CLI

Para descrever conexões

O `describe-connections` exemplo a seguir lista as conexões que você testou entre uma instância de replicação e um endpoint.

```
aws dms describe-connections
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Criação de endpoints de origem e destino](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeConnections](#) na Referência de AWS CLI Comandos.

describe-endpoint-types

O código de exemplo a seguir mostra como usar `describe-endpoint-types`.

AWS CLI

Para listar os tipos de endpoints disponíveis

O `describe-endpoint-types` exemplo a seguir lista os tipos de My SQL endpoint que estão disponíveis.

```
aws dms describe-endpoint-types \
  --filters "Name=engine-name,Values=mysql"
```

Saída:

```
{
  "SupportedEndpointTypes": [
    {
      "EngineName": "mysql",
      "SupportsCDC": true,
      "EndpointType": "source",
      "EngineDisplayName": "MySQL"
    },
    {
      "EngineName": "mysql",
      "SupportsCDC": true,
      "EndpointType": "target",
      "EngineDisplayName": "MySQL"
    }
  ]
}
```

Para obter mais informações, consulte [Working with AWS DMS Endpoints](https://docs.aws.amazon.com/dms/latest/userguide/_endpoints.html) < https://docs.aws.amazon.com/dms/latest/userguide/_endpoints.html > no Guia do usuário CHAP do Database Migration Service.AWS

- Para API obter detalhes, consulte [DescribeEndpointTypes](#) na Referência de AWS CLI Comandos.

describe-endpoints

O código de exemplo a seguir mostra como usar `describe-endpoints`.

AWS CLI

Para descrever os endpoints

O `describe-endpoints` exemplo a seguir lista os endpoints em sua AWS conta.

```
aws dms describe-endpoints
```

Saída:

```
{
  "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
    }
]
}

```

Para obter mais informações, consulte [Trabalhando com AWS DMS endpoints](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeEndpoints](#) na Referência de AWS CLI Comandos.

describe-event-categories

O código de exemplo a seguir mostra como usar `describe-event-categories`.

AWS CLI

Para descrever as categorias de eventos

O `describe-event-categories` exemplo a seguir lista as categorias de eventos disponíveis.

aws dms describe-event-categories

Saída:

```
{
  "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"
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhando com eventos e notificações](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeEventCategories](#) na Referência de AWS CLI Comandos.

describe-event-subscriptions

O código de exemplo a seguir mostra como usar `describe-event-subscriptions`.

AWS CLI

Para descrever as assinaturas de eventos

O `describe-event-subscriptions` exemplo a seguir lista as assinaturas de eventos para um tópico da AmazonSNS.

```
aws dms describe-event-subscriptions
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhando com eventos e notificações](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeEventSubscriptions](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para listar DMS eventos

O `describe-events` exemplo a seguir lista os eventos que se originaram de uma instância de replicação.

```
aws dms describe-events \
```

```
--source-type "replication-instance"
```

Saída:

```
{
  "Events": [
    {
      "SourceIdentifier": "my-repl-instance",
      "SourceType": "replication-instance",
      "Message": "Replication application shutdown",
      "EventCategories": [],
      "Date": 1590771645.776
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhando com eventos e notificações](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-orderable-replication-instances

O código de exemplo a seguir mostra como usar `describe-orderable-replication-instances`.

AWS CLI

Para descrever instâncias de replicação solicitáveis

O `describe-orderable-replication-instances` exemplo a seguir lista os tipos de instância de replicação que você pode solicitar.

```
aws dms describe-orderable-replication-instances
```

Saída:

```
{
  "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...
}
```

Para obter mais informações, consulte Como [trabalhar com uma instância de AWS DMS replicação](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeOrderableReplicationInstances](#) na Referência de AWS CLI Comandos.

describe-refresh-schemas-status

O código de exemplo a seguir mostra como usar `describe-refresh-schemas-status`.

AWS CLI

Para listar o status de atualização de um endpoint

O `describe-refresh-schemas-status` exemplo a seguir retorna o status de uma solicitação de atualização anterior.

```
aws dms describe-refresh-schemas-status \
  --endpoint-arn arn:aws:dms:us-
  east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA
```

Saída:

```
{
  "RefreshSchemasStatus": {
    "EndpointArn": "arn:aws:dms:us-
    east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",
    "ReplicationInstanceArn": "arn:aws:dms:us-
    east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "Status": "successful",
    "LastRefreshDate": 1590786544.605
  }
}
```

- Para API obter detalhes, consulte [DescribeRefreshSchemasStatus](#) na Referência de AWS CLI Comandos.

describe-replication-instances

O código de exemplo a seguir mostra como usar `describe-replication-instances`.

AWS CLI

Para descrever instâncias de replicação

O `describe-replication-instances` exemplo a seguir lista as instâncias de replicação na sua AWS conta.

aws dms describe-replication-instances

Saída:

```
{
  "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
  }
]
}

```

Para obter mais informações, consulte Como [trabalhar com uma instância de AWS DMS replicação](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeReplicationInstances](#) na Referência de AWS CLI Comandos.

describe-replication-subnet-groups

O código de exemplo a seguir mostra como usar describe-replication-subnet-groups.

AWS CLI

Para exibir os grupos de sub-redes disponíveis

O describe-replication-subnet-groups exemplo a seguir lista os grupos de sub-redes disponíveis.

```

aws dms describe-replication-subnet-groups \
  --filter "Name=replication-subnet-group-id,Values=my-subnet-group"

```

Saída:

```

{
  "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"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Configurando uma rede para uma instância de replicação](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeReplicationSubnetGroups](#) na Referência de AWS CLI Comandos.

describe-replication-task-assessment-results

O código de exemplo a seguir mostra como usar `describe-replication-task-assessment-results`.

AWS CLI

Para listar os resultados das avaliações de tarefas de replicação

O `describe-replication-task-assessment-results` exemplo a seguir lista os resultados de uma avaliação anterior da tarefa.

```
aws dms describe-replication-task-assessment-results
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Criação de um relatório de avaliação de tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeReplicationTaskAssessmentResults](#) na Referência de AWS CLI Comandos.

describe-replication-tasks

O código de exemplo a seguir mostra como usar `describe-replication-tasks`.

AWS CLI

Para descrever uma tarefa de replicação

O `describe-replication-tasks` exemplo a seguir descreve as tarefas de replicação atuais.

```
aws dms describe-replication-tasks
```

Saída:

```
{
  "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
    }
  }
]
}

```

Para obter mais informações, consulte [Trabalhando com AWS DMS tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeReplicationTasks](#) na Referência de AWS CLI Comandos.

describe-schemas

O código de exemplo a seguir mostra como usar describe-schemas.

AWS CLI

Para descrever esquemas de banco de dados

O describe-schemas exemplo a seguir lista as tabelas disponíveis em um endpoint.

```
aws dms describe-schemas \
```

```
--endpoint-arn "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA"
```

Saída:

```
{
  "Schemas": [
    "prodrep"
  ]
}
```

Para obter mais informações, consulte [Este é o título do tópico](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [DescribeSchemas](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags de uma instância de replicação.

```
aws dms list-tags-for-resource \
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Saída:

```
{
  "TagList": [
    {
      "Key": "Project",
      "Value": "dbMigration"
    },
    {
      "Key": "Environment",
      "Value": "PROD"
    }
  ]
}
```

```
}

```

Para obter mais informações, consulte [Tagging Resources](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

modify-endpoint

O código de exemplo a seguir mostra como usar `modify-endpoint`.

AWS CLI

Para modificar um endpoint

O `modify-endpoint` exemplo a seguir adiciona um atributo de conexão extra a um endpoint.

```
aws dms modify-endpoint \
  --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U" \
  --extra-connection-attributes "compressionType=GZIP"
```

Saída:

```
{
  "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
    }
}
```

Para obter mais informações, consulte [Working with AWS DMS Endpoints](https://docs.aws.amazon.com/dms/latest/userguide/_endpoints.html) < https://docs.aws.amazon.com/dms/latest/userguide/_endpoints.html >`__ no Guia do usuário CHAP do Database Migration Service.AWS

- Para API obter detalhes, consulte [ModifyEndpoint](#) na Referência de AWS CLI Comandos.

modify-event-subscription

O código de exemplo a seguir mostra como usar `modify-event-subscription`.

AWS CLI

Para modificar a assinatura de um evento

O `modify-event-subscription` exemplo a seguir altera o tipo de origem de uma assinatura de evento.

```
aws dms modify-event-subscription \
  --subscription-name "my-dms-events" \
  --source-type replication-task
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Trabalhando com eventos e notificações](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [ModifyEventSubscription](#) na Referência de AWS CLI Comandos.

modify-replication-instance

O código de exemplo a seguir mostra como usar `modify-replication-instance`.

AWS CLI

Para modificar uma instância de replicação

O `modify-replication-instance` exemplo a seguir modifica uma instância de replicação para que ela use uma implantação Multi-AZ.

```
aws dms modify-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \  
  --multi-az
```

Saída:

```
{  
  "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...
  }
}

```

Para obter mais informações, consulte Como [trabalhar com uma instância de AWS DMS replicação](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [ModifyReplicationInstance](#) na Referência de AWS CLI Comandos.

modify-replication-subnet-group

O código de exemplo a seguir mostra como usar `modify-replication-subnet-group`.

AWS CLI

Para modificar um grupo de sub-redes

O `modify-replication-subnet-group` exemplo a seguir altera as listas de sub-redes associadas a um grupo de sub-redes.

```

aws dms modify-replication-subnet-group \
  --replication-subnet-group-identifier my-subnet-group \
  --subnet-id subnet-da327bf6 subnet-bac383e0

```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Configurando uma rede para uma instância de replicação](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [ModifyReplicationSubnetGroup](#) na Referência de AWS CLI Comandos.

modify-replication-task

O código de exemplo a seguir mostra como usar `modify-replication-task`.

AWS CLI

Para modificar uma tarefa de replicação

O `modify-replication-task` exemplo a seguir altera os mapeamentos de tabela de uma tarefa.

```

aws dms modify-replication-task \
  --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
  --table-mappings file://table-mappings.json

```

Conteúdo de `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": []
    }
]
}

```

Saída:

```

{
  "ReplicationTask": {
    "ReplicationTaskIdentifier": "moveit2",
    "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
    "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "MigrationType": "full-load",
    "TableMappings": ...output omitted...,
    "ReplicationTaskSettings": ...output omitted...,
    "Status": "modifying",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590789424.653,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Para obter mais informações, consulte [Trabalhando com AWS DMS tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [ModifyReplicationTask](#) na Referência de AWS CLI Comandos.

reboot-replication-instance

O código de exemplo a seguir mostra como usar `reboot-replication-instance`.

AWS CLI

Para reinicializar uma instância de replicação

O exemplo da `reboot-replication-instance` a seguir reinicializa uma instância de replicação.

```
aws dms reboot-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Saída:

```
{  
  "ReplicationInstance": {  
    "ReplicationInstanceIdentifier": "my-repl-instance",  
    "ReplicationInstanceClass": "dms.t2.micro",  
    "ReplicationInstanceStatus": "rebooting",  
    "AllocatedStorage": 5,  
    "InstanceCreateTime": 1590011235.952,  
    ... output omitted ...  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com uma instância de AWS DMS replicação](#) no Guia do usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [RebootReplicationInstance](#) na Referência de AWS CLI Comandos.

refresh-schemas

O código de exemplo a seguir mostra como usar `refresh-schemas`.

AWS CLI

Para atualizar esquemas de banco de dados

O `refresh-schemas` exemplo a seguir solicita que AWS DMS atualize a lista de esquemas em um endpoint.

```
aws dms refresh-schemas \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --endpoint ...
```

```
--endpoint-arn "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA"
```

Saída:

```
{
  "RefreshSchemasStatus": {
    "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "Status": "refreshing",
    "LastRefreshDate": 1590019949.103
  }
}
```

- Para API obter detalhes, consulte [RefreshSchemas](#) na Referência de AWS CLI Comandos.

reload-tables

O código de exemplo a seguir mostra como usar `reload-tables`.

AWS CLI

Para atualizar a lista de tabelas disponíveis em um endpoint

O `reload-tables` exemplo a seguir recarrega a lista de tabelas disponíveis em um 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"
```

Saída:

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- Para API obter detalhes, consulte [ReloadTables](#) na Referência de AWS CLI Comandos.

remove-tags-from-resource

O código de exemplo a seguir mostra como usar `remove-tags-from-resource`.

AWS CLI

Para remover tags de uma instância de replicação

O `remove-tags-from-resource` exemplo a seguir remove as tags de uma instância de replicação.

```
aws dms remove-tags-from-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --tag-keys Environment Project
```

Este comando não produz saída.

Para obter mais informações, consulte [Tagging Resources](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [RemoveTagsFromResource](#) na Referência de AWS CLI Comandos.

start-replication-task-assessment

O código de exemplo a seguir mostra como usar `start-replication-task-assessment`.

AWS CLI

Para iniciar uma avaliação de tarefas

O `start-replication-task-assessment` exemplo a seguir inicia uma avaliação da tarefa de replicação.

```
aws dms start-replication-task-assessment \  
  --replication-task-arn arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Saída:

```
{
```

```

"ReplicationTask": {
  "ReplicationTaskIdentifier": "moveit2",
  "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",
  "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:E0M4SFKCZEYHZBFGAGZT3QEC5U",
  "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
  "MigrationType": "full-load",
  "TableMappings": ...output omitted...,
  "ReplicationTaskSettings": ...output omitted...,
  "Status": "testing",
  "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
  "ReplicationTaskCreationDate": 1590524772.505,
  "ReplicationTaskStartDate": 1590789988.677,
  "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
}

```

Para obter mais informações, consulte [Criação de um relatório de avaliação de tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [StartReplicationTaskAssessment](#) na Referência de AWS CLI Comandos.

start-replication-task

O código de exemplo a seguir mostra como usar `start-replication-task`.

AWS CLI

Para iniciar uma tarefa de replicação

O command-name exemplo a seguir lista os widgets disponíveis na sua AWS conta.

```

aws dms start-replication-task \
  --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \
  --start-replication-task-type reload-target

```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Trabalhando com AWS DMS tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [StartReplicationTask](#) na Referência de AWS CLI Comandos.

stop-replication-task

O código de exemplo a seguir mostra como usar stop-replication-task.

AWS CLI

Para interromper uma tarefa

O stop-replication-task exemplo a seguir interrompe uma tarefa.

```
aws dms stop-replication-task \
  --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Saída:

```
{
```



```

"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"
}
}

```

Para obter mais informações, consulte [Trabalhando com AWS DMS tarefas](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [StopReplicationTask](#) na Referência de AWS CLI Comandos.

test-connection

O código de exemplo a seguir mostra como usar `test-connection`.

AWS CLI

Para testar uma conexão com um endpoint

O `test-connection` exemplo a seguir testa se um endpoint pode ser acessado a partir de uma instância de replicação.

```

aws dms test-connection \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \
  --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA

```

Saída:

```
{
  "Connection": {
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
    "Status": "testing",
    "EndpointIdentifier": "src-database-1",
    "ReplicationInstanceIdentifier": "my-repl-instance"
  }
}
```

Para obter mais informações, consulte [Criação de endpoints de origem e destino](#) no Guia do Usuário do AWS Database Migration Service.

- Para API obter detalhes, consulte [TestConnection](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon DocumentDB usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon DocumentDB.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-resource

O código de exemplo a seguir mostra como usar `add-tags-to-resource`.

AWS CLI

Para adicionar uma ou mais tags a um recurso especificado

O `add-tags-to-resource` exemplo a seguir adiciona três tags `sample-cluster` a. Uma tag (`CropB`) tem um nome de chave, mas nenhum valor.

```
aws docdb add-tags-to-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
  --tags Key="CropA",Value="Apple" Key="CropB" Key="CropC",Value="Corn"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar recursos do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [AddTagsToResource](#) na Referência de AWS CLI Comandos.

apply-pending-maintenance-action

O código de exemplo a seguir mostra como usar `apply-pending-maintenance-action`.

AWS CLI

Para que as ações de manutenção pendentes ocorram durante a próxima janela de manutenção

O `apply-pending-maintenance-action` exemplo a seguir faz com que todas as ações de atualização do sistema sejam executadas durante a próxima janela de manutenção programada.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Aplicação de atualizações do Amazon DocumentDB no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ApplyPendingMaintenanceAction](#) na Referência de AWS CLI Comandos.

copy-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `copy-db-cluster-parameter-group`.

AWS CLI

Para duplicar um grupo de parâmetros de cluster de banco de dados existente

O `copy-db-cluster-parameter-group` exemplo a seguir faz uma cópia do grupo de parâmetros `custom-docdb3-6` chamado `custom-docdb3-6-copy`. Ao fazer a cópia, ele adiciona tags ao novo grupo de parâmetros.

```
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"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Cópia de um grupo de parâmetros de cluster do Amazon DocumentDB no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [CopyDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

`copy-db-cluster-snapshot`

O código de exemplo a seguir mostra como usar `copy-db-cluster-snapshot`.

AWS CLI

Para criar uma cópia de um snapshot

O exemplo da `copy-db-cluster-snapshot` a seguir faz uma cópia do `sample-cluster-snapshot` denominado `sample-cluster-snapshot-copy`. A cópia tem todas as tags do original, além de uma nova tag com o nome da chave `CopyNumber`.

```
aws docdb copy-db-cluster-snapshot \  
  --source-db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \  
  --copy-tags \  
  --tags Key="CopyNumber",Value="1"
```

Este comando não produz saída.

Para obter mais informações, consulte [Cópia de um snapshot de cluster](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [CopyDbClusterSnapshot](#) na Referência de AWS CLI Comandos.

create-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `create-db-cluster-parameter-group`.

AWS CLI

Para criar um grupo de parâmetros de cluster do Amazon DocumentDB

O `create-db-cluster-parameter-group` exemplo a seguir cria o grupo de parâmetros do cluster de banco de dados `sample-parameter-group` usando a `docdb3.6` família.

```
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"
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Criação de um grupo de parâmetros de cluster do Amazon DocumentDB no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [CreateDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

create-db-cluster-snapshot

O código de exemplo a seguir mostra como usar `create-db-cluster-snapshot`.

AWS CLI

Para criar um snapshot manual do cluster Amazon DocumentDB

O `create-db-cluster-snapshot` exemplo a seguir cria um snapshot de cluster de banco de dados da Amazon chamado `sample-cluster-snapshot`.

```

aws docdb create-db-cluster-snapshot \
  --db-cluster-identifier sample-cluster \
  --db-cluster-snapshot-identifier sample-cluster-snapshot

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Criação manual de um snapshot de cluster](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [CreateDbClusterSnapshot](#) na Referência de AWS CLI Comandos.

create-db-cluster

O código de exemplo a seguir mostra como usar `create-db-cluster`.

AWS CLI

Para criar um cluster Amazon DocumentDB

O `create-db-cluster` exemplo a seguir cria um cluster Amazon DocumentDB chamado `sample-cluster` com a janela de manutenção preferencial aos domingos, entre 20:30 e 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

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Criação de um cluster Amazon DocumentDB no Guia do desenvolvedor do Amazon DocumentDB](#).

- Para API obter detalhes, consulte [CreateDbCluster](#) na Referência de AWS CLI Comandos.

create-db-instance

O código de exemplo a seguir mostra como usar create-db-instance.

AWS CLI

Para criar uma instância de cluster do Amazon DocumentDB

O código de `create-db-instance` exemplo a seguir cria a instância `sample-cluster-instance-2` no 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
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Adicionar uma instância do Amazon DocumentDB a um cluster](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [CreateDbInstance](#) na Referência de AWS CLI Comandos.

create-db-subnet-group

O código de exemplo a seguir mostra como usar create-db-subnet-group.

AWS CLI

Para criar um grupo de sub-redes do Amazon DocumentDB

O create-db-subnet-group exemplo a seguir cria um grupo de sub-rede Amazon DocumentDB chamado. 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"
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Criação de um grupo de sub-rede do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [CreateDbSubnetGroup](#) na Referência de AWS CLI Comandos.

delete-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `delete-db-cluster-parameter-group`.

AWS CLI

Para excluir um grupo de parâmetros de cluster do Amazon DocumentDB

O `delete-db-cluster-parameter-group` exemplo a seguir exclui o grupo de parâmetros do Amazon DocumentDB. `sample-parameter-group`

```
aws docdb delete-db-cluster-parameter-group \
  --db-cluster-parameter-group-name sample-parameter-group
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um grupo de parâmetros de cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DeleteDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

delete-db-cluster-snapshot

O código de exemplo a seguir mostra como usar `delete-db-cluster-snapshot`.

AWS CLI

Para excluir um snapshot de cluster do Amazon DocumentDB

O `delete-db-cluster-snapshot` exemplo a seguir exclui o snapshot do cluster Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Excluindo um snapshot de cluster](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DeleteDbClusterSnapshot](#) na Referência de AWS CLI Comandos.

delete-db-cluster

O código de exemplo a seguir mostra como usar `delete-db-cluster`.

AWS CLI

Para excluir um cluster do Amazon DocumentDB

O `delete-db-cluster` exemplo a seguir exclui o cluster Amazon `sample-cluster` DocumentDB. Nenhum backup do cluster é feito antes de excluí-lo. NOTE: você deve excluir todas as instâncias associadas ao cluster antes de poder excluí-lo.

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --skip-final-snapshot
```

Saída:

```
{  
  "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": []
}
}

```

Para obter mais informações, consulte [Excluir um cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DeleteDbCluster](#) na Referência de AWS CLI Comandos.

delete-db-instance

O código de exemplo a seguir mostra como usar delete-db-instance.

AWS CLI

Para excluir uma instância do Amazon DocumentDB

O delete-db-instance exemplo a seguir exclui a instância do Amazon sample-cluster-instance-2 DocumentDB.

```
aws docdb delete-db-instance \
  --db-instance-identifier sample-cluster-instance-2
```

Saída:

```
{
  "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": "ZNKXH85TT8WW",
      "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
  }
}

```

Para obter mais informações, consulte [Excluir uma instância do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DeleteDbInstance](#) na Referência de AWS CLI Comandos.

delete-db-subnet-group

O código de exemplo a seguir mostra como usar delete-db-subnet-group.

AWS CLI

Para excluir um grupo de sub-redes do Amazon DocumentDB

O delete-db-subnet-group exemplo a seguir exclui o grupo de sub-rede Amazon DocumentDB. sample-subnet-group

```

aws docdb delete-db-subnet-group \
  --db-subnet-group-name sample-subnet-group

```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um grupo de sub-rede do Amazon DocumentDB no Guia do desenvolvedor](#) do Amazon DocumentDB.

- Para API obter detalhes, consulte [DeleteDbSubnetGroup](#) na Referência de AWS CLI Comandos.

describe-db-cluster-parameter-groups

O código de exemplo a seguir mostra como usar `describe-db-cluster-parameter-groups`.

AWS CLI

Para ver os detalhes de um ou mais grupos de parâmetros de cluster do Amazon DocumentDB

O `describe-db-cluster-parameter-groups` exemplo a seguir exibe detalhes do grupo de parâmetros do cluster Amazon DocumentDB. `custom3-6-param-grp`

```
aws docdb describe-db-cluster-parameter-groups \
  --db-cluster-parameter-group-name custom3-6-param-grp
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Visualização dos grupos de parâmetros de cluster do Amazon DocumentDB no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbClusterParameterGroups](#) na Referência de AWS CLI Comandos.

describe-db-cluster-parameters

O código de exemplo a seguir mostra como usar `describe-db-cluster-parameters`.

AWS CLI

Para visualizar a lista detalhada de parâmetros de um grupo de parâmetros do cluster Amazon DocumentDB.

O `describe-db-cluster-parameters` exemplo a seguir lista os parâmetros do grupo de parâmetros `custom3-6-param-grp` do Amazon DocumentDB.

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name custom3-6-param-grp
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Visualização dos parâmetros de cluster do Amazon DocumentDB no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbClusterParameters](#) na Referência de AWS CLI Comandos.

describe-db-cluster-snapshot-attributes

O código de exemplo a seguir mostra como usar `describe-db-cluster-snapshot-attributes`.

AWS CLI

Para listar os nomes e valores de atributos de um snapshot do Amazon DocumentDB

O `describe-db-cluster-snapshot-attributes` exemplo a seguir lista os nomes e valores dos atributos para o snapshot do Amazon DocumentDB. `sample-cluster-snapshot`

```

aws docdb describe-db-cluster-snapshot-attributes \
  --db-cluster-snapshot-identifier sample-cluster-snapshot

```

Saída:

```

{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": []
      }
    ]
  }
}

```

```
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"  
  }  
}
```

Para obter mais informações, consulte [D `describeDBClusterSnapshotAttributes`](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbClusterSnapshotAttributes](#) na Referência de AWS CLI Comandos.

`describe-db-cluster-snapshots`

O código de exemplo a seguir mostra como usar `describe-db-cluster-snapshots`.

AWS CLI

Para descrever os snapshots do Amazon DocumentDB

O `describe-db-cluster-snapshots` exemplo a seguir exibe detalhes do snapshot do Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb describe-db-cluster-snapshots \  
--db-cluster-snapshot-identifier sample-cluster-snapshot
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [DescrevaDBCluster Snapshots no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbClusterSnapshots](#) na Referência de AWS CLI Comandos.

describe-db-clusters

O código de exemplo a seguir mostra como usar `describe-db-clusters`.

AWS CLI

Para obter informações detalhadas sobre um ou mais clusters do Amazon DocumentDB.

O `describe-db-clusters` exemplo a seguir exibe detalhes do cluster Amazon DocumentDB. `sample-cluster` Ao omitir o `--db-cluster-identifier` parâmetro, você pode obter informações de até 100 clusters.

```

aws docdb describe-db-clusters
--db-cluster-identifier sample-cluster

```

Saída:

```

{
  "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": "ZNKXH85TT8WWW",
"StorageEncrypted": false,
"EnabledCloudwatchLogsExports": [
  "audit"
```

```
    ],
    "AssociatedRoles": [],
    "EngineVersion": "3.6.0",
    "Port": 27017,
    "Status": "available"
  }
]
```

Para obter mais informações, consulte [Descrição dos clusters do Amazon DocumentDB no Guia do desenvolvedor do Amazon DocumentDB](#).

- Para API obter detalhes, consulte [DescribeDbClusters](#) na Referência de AWS CLI Comandos.

describe-db-engine-versions

O código de exemplo a seguir mostra como usar `describe-db-engine-versions`.

AWS CLI

Para listar as versões disponíveis do mecanismo Amazon DocumentDB

O `describe-db-engine-versions` exemplo a seguir lista todas as versões disponíveis do mecanismo Amazon DocumentDB.

```
aws docdb describe-db-engine-versions \
  --engine docdb
```

Saída:

```
{
  "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"
      ]
    }
  ]
}
```



```

    }
  ]
}

```

Para obter mais informações, consulte [escribeDBEngineVersões D](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbEngineVersions](#) na Referência de AWS CLI Comandos.

describe-db-instances

O código de exemplo a seguir mostra como usar `describe-db-instances`.

AWS CLI

Para encontrar informações sobre instâncias provisionadas do Amazon DocumentDB

O `describe-db-instances` exemplo a seguir exibe detalhes sobre a instância do Amazon DocumentDB. `sample-cluster-instance` Ao omitir o `--db-instance-identifier` parâmetro, você obtém informações sobre até 100 instâncias.

```
aws docdb describe-db-instances \
  --db-instance-identifier sample-cluster-instance
```

Saída:

```
{
  "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-A2GIKUV6KP0HITGGKI2NHVISZA"
  }
]
}

```

Para obter mais informações, consulte [Descrição das instâncias do Amazon DocumentDB no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbInstances](#) na Referência de AWS CLI Comandos.

describe-db-subnet-groups

O código de exemplo a seguir mostra como usar `describe-db-subnet-groups`.

AWS CLI

Para recuperar uma lista de descrições de sub-rede do Amazon DocumentDB

O `describe-db-subnet-groups` exemplo a seguir descreve os detalhes da sub-rede Amazon DocumentDB chamada `default`

```

aws docdb describe-db-subnet-groups \
  --db-subnet-group-name default

```

Saída:

```

{
  "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"
}
]
```

Para obter mais informações, consulte [Descrevendo grupos de sub-redes](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeDbSubnetGroups](#) na Referência de AWS CLI Comandos.

describe-engine-default-cluster-parameters

O código de exemplo a seguir mostra como usar `describe-engine-default-cluster-parameters`.

AWS CLI

Para descrever o mecanismo padrão e as informações de parâmetros do sistema para o Amazon DocumentDB

O `describe-engine-default-cluster-parameters` exemplo a seguir exibe detalhes do mecanismo padrão e das informações de parâmetros do sistema para o grupo de parâmetros do Amazon DocumentDB. `docdb3.6`

```
aws docdb describe-engine-default-cluster-parameters \
  --db-parameter-group-family docdb3.6
```

Saída:

```
{
  "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
    }
]
}
}

```

Para obter mais informações, consulte o [DescribeEngineDefaultClusterParameters](#) Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeEngineDefaultClusterParameters](#) na Referência de AWS CLI Comandos.

describe-event-categories

O código de exemplo a seguir mostra como usar `describe-event-categories`.

AWS CLI

Para descrever todas as categorias de eventos do Amazon DocumentDB

O `describe-event-categories` exemplo a seguir lista todas as categorias do tipo de fonte de eventos Amazon DocumentDB. `db-instance`

```
aws docdb describe-event-categories \
  --source-type db-cluster
```

Saída:

```
{
  "EventCategoriesMapList": [
    {
      "SourceType": "db-cluster",
```

```

        "EventCategories": [
            "failover",
            "maintenance",
            "notification",
            "failure"
        ]
    }
]
}

```

Para obter mais informações, consulte [Visualização de categorias de eventos](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeEventCategories](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para listar eventos do Amazon DocumentDB

O `describe-events` exemplo a seguir lista todos os eventos do Amazon DocumentDB nas últimas 24 horas (1440 minutos).

```
aws docdb describe-events \
  --duration 1440
```

Este comando não produz saída. Saída:

```

{
  "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"
  }
}
```

```
]
}
```

Para obter mais informações, consulte [Visualização de eventos do Amazon DocumentDB no Guia do desenvolvedor do Amazon DocumentDB](#).

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-orderable-db-instance-options

O código de exemplo a seguir mostra como usar `describe-orderable-db-instance-options`.

AWS CLI

Para encontrar as opções de instância do Amazon DocumentDB, você pode solicitar

O `describe-orderable-db-instance-options` exemplo a seguir lista todas as opções de instância do Amazon DocumentDB para uma região.

```
aws docdb describe-orderable-db-instance-options \
  --engine docdb \
  --region us-east-1
```

Saída:

```
{
  "OrderableDBInstanceOptions": [
    {
      "Vpc": true,
      "AvailabilityZones": [
        {
          "Name": "us-east-1a"
        },
        {
          "Name": "us-east-1b"
        },
        {
          "Name": "us-east-1c"
        },
        {
          "Name": "us-east-1d"
        }
      ]
    }
  ],
}
```

```
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.16xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.2xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
```

```
    "DBInstanceClass": "db.r4.4xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.8xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.large",
    "LicenseModel": "na",
```

```
    "Engine": "docdb"
  },
  {
    "Vpc": true,
    "AvailabilityZones": [
      {
        "Name": "us-east-1a"
      },
      {
        "Name": "us-east-1b"
      },
      {
        "Name": "us-east-1c"
      },
      {
        "Name": "us-east-1d"
      }
    ],
    "EngineVersion": "3.6.0",
    "DBInstanceClass": "db.r4.xlarge",
    "LicenseModel": "na",
    "Engine": "docdb"
  }
]
```

Para obter mais informações, consulte [Adicionar uma instância do Amazon DocumentDB a um cluster](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [DescribeOrderableDbInstanceOptions](#) na Referência de AWS CLI Comandos.

describe-pending-maintenance-actions

O código de exemplo a seguir mostra como usar `describe-pending-maintenance-actions`.

AWS CLI

Para listar suas ações pendentes de manutenção do Amazon DocumentDB

O `describe-pending-maintenance-actions` exemplo a seguir lista todas as suas ações pendentes de manutenção do Amazon DocumentDB.

```
aws docdb describe-pending-maintenance-actions
```

Saída:

```
{
  "PendingMaintenanceActions": []
}
```

Para obter mais informações, consulte [Manutenção do Amazon DocumentDB no Guia do desenvolvedor do Amazon DocumentDB](#).

- Para API obter detalhes, consulte [DescribePendingMaintenanceActions](#) na Referência de AWS CLI Comandos.

failover-db-cluster

O código de exemplo a seguir mostra como usar `failover-db-cluster`.

AWS CLI

Para forçar o failover de um cluster do Amazon DocumentDB para uma réplica

O `failover-db-cluster` exemplo a seguir faz com que a instância primária no cluster de amostra do cluster Amazon DocumentDB passe por failover para uma réplica.

```
aws docdb failover-db-cluster \
  --db-cluster-identifier sample-cluster
```

Saída:

```
{
  "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": "ZNKXH85TT8WWW",
  "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"
}
```



```
}
```

Para obter mais informações, consulte [Amazon DocumentDB Failover](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [FailoverDbCluster](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar todas as tags em um recurso do Amazon DocumentDB

O `list-tags-for-resource` exemplo a seguir lista todas as tags no cluster Amazon DocumentDB. `sample-cluster`

```
aws docdb list-tags-for-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster
```

Saída:

```
{  
  "TagList": [  
    {  
      "Key": "A",  
      "Value": "ALPHA"  
    },  
    {  
      "Key": "B",  
      "Value": ""  
    },  
    {  
      "Key": "C",  
      "Value": "CHARLIE"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Listar tags em um recurso do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

modify-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `modify-db-cluster-parameter-group`.

AWS CLI

Para modificar um grupo de parâmetros do cluster de banco de dados Amazon DocumentDB

O `modify-db-cluster-parameter-group` exemplo a seguir modifica o `custom3-6-param-grp` grupo de parâmetros do cluster Amazon DocumentDB definindo os dois `audit_logs` parâmetros `ttl_monitor` e como habilitado. As alterações serão aplicadas na próxima reinicialização.

```
aws docdb modify-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --  
  parameters ParameterName=audit_logs,ParameterValue=enabled,ApplyMethod=pending-reboot \  
  
  ParameterName=ttl_monitor,ParameterValue=enabled,ApplyMethod=pending-reboot
```

Saída:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Para obter mais informações, consulte [Modificação de um grupo de parâmetros de cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ModifyDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

modify-db-cluster-snapshot-attribute

O código de exemplo a seguir mostra como usar `modify-db-cluster-snapshot-attribute`.

AWS CLI

Exemplo 1: Para adicionar um atributo a um snapshot do Amazon DocumentDB

O `modify-db-cluster-snapshot-attribute` exemplo a seguir adiciona quatro valores de atributos a um snapshot do 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
```

Saída:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789011",  
          "123456789012",  
          "123456789013"  
        ]  
      }  
    ],  
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"  
  }  
}
```

Exemplo 2: Para remover atributos de um snapshot do Amazon DocumentDB

O `modify-db-cluster-snapshot-attribute` exemplo a seguir remove dois valores de atributos de um snapshot do cluster Amazon DocumentDB.

```
aws docdb modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --attribute-name restore \  
  --values-to-remove 123456789012
```

Saída:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789011",
          "123456789013"
        ]
      }
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
  }
}
```

Para obter mais informações, consulte [ModifyDBCluster SnapshotAttribute](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ModifyDbClusterSnapshotAttribute](#) na Referência de AWS CLI Comandos.

modify-db-cluster

O código de exemplo a seguir mostra como usar `modify-db-cluster`.

AWS CLI

Para modificar um cluster do Amazon DocumentDB

O `modify-db-cluster` exemplo a seguir modifica o `sample-cluster` cluster Amazon DocumentDB fazendo com que o período de retenção dos backups automáticos seja de 7 dias e alterando as janelas preferenciais para backups e manutenção. Todas as alterações são aplicadas na próxima janela de manutenção.

```
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
```

Saída:

```
{
  "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
  }
}

```

Para obter mais informações, consulte [Modificação de um cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ModifyDbCluster](#) na Referência de AWS CLI Comandos.

modify-db-instance

O código de exemplo a seguir mostra como usar `modify-db-instance`.

AWS CLI

Para modificar uma instância do Amazon DocumentDB

O `modify-db-instance` exemplo a seguir modifica a `sample-cluster2` instância do Amazon DocumentDB alterando sua classe de instância `db.r4.4xlarge` para e seu nível de promoção para `5`. As alterações são aplicadas imediatamente, mas só podem ser vistas depois que o status da instância estiver disponível.

```

aws docdb modify-db-instance \
  --db-instance-identifier sample-cluster2 \
  --apply-immediately \
  --db-instance-class db.r4.4xlarge \
  --promotion-tier 5

```

Saída:

```

{
  "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": "ZNKXH85TT8WWV",
    "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"
}
}
```

Para obter mais informações, consulte [Modificar uma instância do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ModifyDbInstance](#) na Referência de AWS CLI Comandos.

modify-db-subnet-group

O código de exemplo a seguir mostra como usar `modify-db-subnet-group`.

AWS CLI

Para modificar um grupo de sub-redes do Amazon DocumentDB

O `modify-db-subnet-group` exemplo a seguir modifica o grupo de sub-redes `sample-subnet-group` adicionando as sub-redes especificadas e uma nova descrição.

```
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"
```

Saída:

```
{  
  "DBSubnetGroup": {  
    "DBSubnetGroupName": "sample-subnet-group",  
    "SubnetGroupStatus": "Complete",  
    "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-  
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"  
        }  
      }  
    ]  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Modificação de um grupo de sub-redes do Amazon DocumentDB no Guia do desenvolvedor](#) do Amazon DocumentDB.

- Para API obter detalhes, consulte [ModifyDbSubnetGroup](#) na Referência de AWS CLI Comandos.

reboot-db-instance

O código de exemplo a seguir mostra como usar `reboot-db-instance`.

AWS CLI

Para reinicializar uma instância do Amazon DocumentDB

O `reboot-db-instance` exemplo a seguir reinicia a instância do Amazon `sample-cluster2` DocumentDB.

```
aws docdb reboot-db-instance \  
  --db-instance-identifier sample-cluster2
```

Este comando não produz saída. Saída:

```
{  
  "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": "ZNKXH85TT8WWV",
  "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"
}
}
```

Para obter mais informações, consulte [Reinicializando um Amazon Instance DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [RebootDbInstance](#) na Referência de AWS CLI Comandos.

remove-tags-from-resource

O código de exemplo a seguir mostra como usar `remove-tags-from-resource`.

AWS CLI

Para remover tags de um recurso do Amazon DocumentDB

O `remove-tags-from-resource` exemplo a seguir remove a tag com a chave nomeada B do 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Removendo tags de um Amazon DocumentDB Resource](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [RemoveTagsFromResource](#) na Referência de AWS CLI Comandos.

reset-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `reset-db-cluster-parameter-group`.

AWS CLI

Para redefinir o valor do parâmetro especificado para seus padrões em um grupo de parâmetros do Amazon DocumentDB

O `reset-db-cluster-parameter-group` exemplo a seguir redefine o parâmetro `tTL_monitor` no `custom3-6-param-grp` grupo de parâmetros Amazon DocumentDB para seu valor padrão.

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters ParameterName=tTL_monitor,ApplyMethod=immediate
```

Saída:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Para obter mais informações, consulte o título no Guia do desenvolvedor do Amazon DocumentDB.

Para redefinir os valores especificados ou todos os parâmetros para seus padrões em um grupo de parâmetros do Amazon DocumentDB

O `reset-db-cluster-parameter-group` exemplo a seguir redefine todos os parâmetros no `custom3-6-param-grp` grupo de parâmetros Amazon DocumentDB para seu valor padrão.

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --reset-all-parameters
```

Saída:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Para obter mais informações, consulte [Redefinição de um grupo de parâmetros de cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [ResetDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

restore-db-cluster-from-snapshot

O código de exemplo a seguir mostra como usar `restore-db-cluster-from-snapshot`.

AWS CLI

Para restaurar um cluster do Amazon DocumentDB a partir de um snapshot automático ou manual

O `restore-db-cluster-from-snapshot` exemplo a seguir cria um novo cluster Amazon DocumentDB chamado `sample-cluster-2019-03-16-00-01-restored` partir do snapshot `aws: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 aws:sample-cluster-2019-03-16-00-01
```

Saída:

```
{  
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Restauração a partir de um snapshot de cluster no Guia do desenvolvedor do Amazon DocumentDB](#).

- Para API obter detalhes, consulte [RestoreDbClusterFromSnapshot](#) na Referência de AWS CLI Comandos.

restore-db-cluster-to-point-in-time

O código de exemplo a seguir mostra como usar `restore-db-cluster-to-point-in-time`.

AWS CLI

Para restaurar um cluster Amazon DocumentDB para um a point-in-time partir de um snapshot manual

O `restore-db-cluster-to-point-in-time` exemplo a seguir usa o `sample-cluster-snapshot` para criar um novo cluster Amazon DocumentDB, `sample-cluster-pit`, usando o tempo restaurável mais 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
```

Saída:

```
{
  "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"
  }
}
```



```
}  
}
```

Para obter mais informações, consulte [Restauração de um snapshot em um ponto no tempo no Guia](#) do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [RestoreDbClusterToPointInTime](#) na Referência de AWS CLI Comandos.

start-db-cluster

O código de exemplo a seguir mostra como usar `start-db-cluster`.

AWS CLI

Para iniciar um cluster Amazon DocumentDB parado

O `start-db-cluster` exemplo a seguir inicia o cluster Amazon DocumentDB especificado.

```
aws docdb start-db-cluster \  
  --db-cluster-identifier sample-cluster
```

Saída:

```
{  
  "DBCluster": {  
    "ClusterCreateTime": "2019-03-19T18:45:01.857Z",  
    "HostedZoneId": "ZNKXH85TT8WVW",  
    "Engine": "docdb",  
    "DBClusterMembers": [],  
    "MultiAZ": false,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1c",  
      "us-east-1f"  
    ],  
    "StorageEncrypted": false,  
    "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-  
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",  
    "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-  
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",  
    "Port": 27017,  
  }  
}
```

```

    "PreferredBackupWindow": "00:00-00:30",
    "DBSubnetGroup": "default",
    "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
    "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
    "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
    "DBClusterParameterGroup": "default.docdb3.6",
    "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
    "MasterUsername": "master-user",
    "EngineVersion": "3.6.0",
    "BackupRetentionPeriod": 3,
    "AssociatedRoles": [],
    "Status": "creating",
    "VpcSecurityGroups": [
      {
        "Status": "active",
        "VpcSecurityGroupId": "sg-77186e0d"
      }
    ]
  }
}

```

Para obter mais informações, consulte Como [interromper e iniciar um cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [StartDbCluster](#) na Referência de AWS CLI Comandos.

stop-db-cluster

O código de exemplo a seguir mostra como usar `stop-db-cluster`.

AWS CLI

Para interromper a execução de um cluster Amazon DocumentDB

O `stop-db-cluster` exemplo a seguir interrompe o cluster Amazon DocumentDB especificado.

```

aws docdb stop-db-cluster \
  --db-cluster-identifier sample-cluster

```

Saída:

```
{
```

```

"DBCluster": {
  "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
  "HostedZoneId": "ZNKXH85TT8WVW",
  "Engine": "docdb",
  "DBClusterMembers": [],
  "MultiAZ": false,
  "AvailabilityZones": [
    "us-east-1a",
    "us-east-1c",
    "us-east-1f"
  ],
  "StorageEncrypted": false,
  "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",
  "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjorzrlsfc.us-east-1.docdb.amazonaws.com",
  "Port": 27017,
  "PreferredBackupWindow": "00:00-00:30",
  "DBSubnetGroup": "default",
  "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
  "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
  "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
  "DBClusterParameterGroup": "default.docdb3.6",
  "DbClusterResourceId": "cluster-X0046Q3RH4LWSYNH3NMZKXPISU",
  "MasterUsername": "master-user",
  "EngineVersion": "3.6.0",
  "BackupRetentionPeriod": 3,
  "AssociatedRoles": [],
  "Status": "creating",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-77186e0d"
    }
  ]
}
}

```

Para obter mais informações, consulte Como [interromper e iniciar um cluster do Amazon DocumentDB](#) no Guia do desenvolvedor do Amazon DocumentDB.

- Para API obter detalhes, consulte [StopDbCluster](#) na Referência de AWS CLI Comandos.

Exemplos do DynamoDB usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o DynamoDB.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-get-item

O código de exemplo a seguir mostra como usar `batch-get-item`.

AWS CLI

Como recuperar vários itens de uma tabela

O exemplo de `batch-get-items` a seguir lê vários itens da tabela `MusicCollection` usando um lote de três solicitações `GetItem` e solicita o número de unidades de capacidade de leitura consumidas pela operação. O comando retorna somente o atributo `AlbumTitle`.

```
aws dynamodb batch-get-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity TOTAL
```

Conteúdo de `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"
}
}

```

Saída:

```

{
  "Responses": {
    "MusicCollection": [
      {
        "AlbumTitle": {
          "S": "Somewhat Famous"
        }
      },
      {
        "AlbumTitle": {
          "S": "Blue Sky Blues"
        }
      },
      {
        "AlbumTitle": {
          "S": "Louder Than Ever"
        }
      }
    ]
  },
  "UnprocessedKeys": {},
  "ConsumedCapacity": [
    {
      "TableName": "MusicCollection",
      "CapacityUnits": 1.5
    }
  ]
}

```

```
]
}
```

Para obter mais informações, consulte [Operações em lote](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [BatchGetItem](#) na Referência de AWS CLI Comandos.

batch-write-item

O código de exemplo a seguir mostra como usar `batch-write-item`.

AWS CLI

Como adicionar vários itens a uma tabela

O exemplo de `batch-write-item` a seguir adiciona três novos itens à tabela `MusicCollection` usando um lote de três solicitações `PutItem`. Ele também solicita informações sobre o número de unidades de capacidade de gravação consumidas pela operação e quaisquer coleções de itens modificadas pela operação.

```
aws dynamodb batch-write-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity INDEXES \  
  --return-item-collection-metrics SIZE
```

Conteúdo de `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"}
        }
    }
}
]
}

```

Saída:

```

{
  "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
      }
    }
  }
]
}

```

Para obter mais informações, consulte [Operações em lote](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [BatchWriteItem](#) na Referência de AWS CLI Comandos.

create-backup

O código de exemplo a seguir mostra como usar create-backup.

AWS CLI

Para criar um backup para uma tabela existente do DynamoDB

O create-backup exemplo a seguir cria um backup da MusicCollection tabela.

```

aws dynamodb create-backup \
  --table-name MusicCollection \
  --backup-name MusicCollectionBackup

```

Saída:

```
{
```



```

    "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
    }
  }
}

```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [CreateBackup](#) na Referência de AWS CLI Comandos.

create-global-table

O código de exemplo a seguir mostra como usar `create-global-table`.

AWS CLI

Para criar uma tabela global

O `create-global-table` exemplo a seguir cria uma tabela global a partir de duas tabelas idênticas nas AWS regiões separadas e especificadas.

```

aws dynamodb create-global-table \
  --global-table-name MusicCollection \
  --replication-group RegionName=us-east-2 RegionName=us-east-1 \
  --region us-east-2

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [CreateGlobalTable](#) na Referência de AWS CLI Comandos.

create-table

O código de exemplo a seguir mostra como usar `create-table`.

AWS CLI

Exemplo 1: como criar uma tabela com tags

O exemplo de `create-table` a seguir usa os atributos especificados e o esquema de chaves para criar uma tabela chamada `MusicCollection`. Essa tabela usa taxa de transferência provisionada e é criptografada em repouso usando o padrão de propriedade. AWS CMK O comando também aplica uma tag à tabela, com uma chave `Owner` e valor de `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

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como criar uma tabela no modo sob demanda

O exemplo a seguir cria uma tabela chamada `MusicCollection` usando o modo sob demanda, em vez do modo de throughput provisionado. Esse método é útil para tabelas com workloads imprevisíveis.

```
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
```

Saída:

```
{  
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 3: Para criar uma tabela e criptografá-la com um Customer Managed CMK

O exemplo a seguir cria uma tabela chamada `MusicCollection` e a criptografa usando uma tabela gerenciada pelo CMK cliente.

```

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

```

Saída:

```

{
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 4: como criar uma tabela com um índice secundário local

O exemplo a seguir usa os atributos especificados e o esquema de chaves para criar uma tabela chamada `MusicCollection` com um índice secundário local chamado `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\"]
      }
    }
  ]"

```

Saída:

```

{
  "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"
    }
]
}
}
```


Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 5: como criar uma tabela com um índice secundário global

O exemplo a seguir cria uma tabela chamada `GameScores` com um índice secundário global chamado `GameTitleIndex`. A tabela base tem uma chave de partição de `UserId` e uma chave de classificação de `GameTitle`, permitindo que você encontre a melhor pontuação de um usuário individual para um jogo específico de forma eficiente, enquanto GSI tem uma chave de partição de `GameTitle` e uma chave de classificação de `TopScore`, permitindo que você encontre rapidamente a maior pontuação geral de um determinado jogo.

```
aws dynamodb create-table \
  --table-name GameScores \
  --attribute-
definitions AttributeName=UserId,AttributeType=S AttributeName=GameTitle,AttributeType=S Att
  \
  --key-schema AttributeName=UserId,KeyType=HASH \
                AttributeName=GameTitle,KeyType=RANGE \
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=5 \
  --global-secondary-indexes \
    "[
      {
        \"IndexName\": \"GameTitleIndex\",
        \"KeySchema\": [
          {\"AttributeName\": \"GameTitle\", \"KeyType\": \"HASH\"},
          {\"AttributeName\": \"TopScore\", \"KeyType\": \"RANGE\"}
        ],
        \"Projection\": {
          \"ProjectionType\": \"INCLUDE\",
          \"NonKeyAttributes\": [\"UserId\"]
        },
        \"ProvisionedThroughput\": {
          \"ReadCapacityUnits\": 10,
          \"WriteCapacityUnits\": 5
        }
      }
    ]"
```

Saída:

```
{
```

```
"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"
    }
]
}
}

```

Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 6: como criar uma tabela com vários índices secundários globais ao mesmo tempo

O exemplo a seguir cria uma tabela chamada GameScores com dois índices secundários globais. Os GSI esquemas são passados por meio de um arquivo, em vez de na linha de 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
```

Conteúdo de `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
    }
  }
]
```

```

    }
  }
]

```

Saída:

```

{
  "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"
    }
]
}
}

```

Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 7: como criar uma tabela com o Streams habilitado

O exemplo a seguir cria uma tabela chamada GameScores com o DynamoDB Streams habilitado. Imagens novas e antigas de cada item serão gravadas no fluxo.

```

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

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Operações básicas nas tabelas](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 8: como criar uma tabela com o fluxo somente de chaves habilitado

O exemplo a seguir cria uma tabela chamada GameScores com o DynamoDB Streams habilitado. Somente os atributos-chave dos itens modificados são gravados no fluxo.

```
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
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

```

        "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
}
}

```

Para obter mais informações, consulte [Captura de dados de alterações com o Amazon DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 9: como criar uma tabela com a classe Standard-Infrequent Access

O exemplo a seguir cria uma tabela chamada GameScores e atribui a classe de tabela Standard-Infrequent Access (DynamoDB Standard-IA). Essa classe de tabela é otimizada para que o armazenamento seja o custo dominante.

```

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

```

Saída:

```

{
  "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
}
}
```

Para obter mais informações, consulte [Classes de tabela](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 10: como criar uma tabela com a proteção contra exclusão habilitada

O exemplo a seguir cria uma tabela chamada GameScores e habilita a proteção contra exclusão.

```
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
```

Saída:

```
{  
  "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
  }
}

```

Para obter mais informações, consulte [Usar a proteção contra exclusão](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [CreateTable](#) na Referência de AWS CLI Comandos.

delete-backup

O código de exemplo a seguir mostra como usar delete-backup.

AWS CLI

Para excluir um backup existente do DynamoDB

O delete-backup exemplo a seguir exclui o backup existente especificado.

```

aws dynamodb delete-backup \
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a

```

Saída:

```

{
  "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": {}
}
```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [DeleteBackup](#) na Referência de AWS CLI Comandos.

delete-item

O código de exemplo a seguir mostra como usar `delete-item`.

AWS CLI

Exemplo 1: como excluir um item

O exemplo de `delete-item` a seguir exclui um item da tabela `MusicCollection` e solicita detalhes sobre o item excluído e a capacidade usada pela solicitação.

```
aws dynamodb delete-item \
  --table-name MusicCollection \
```

```
--key file://key.json \  
--return-values ALL_OLD \  
--return-consumed-capacity TOTAL \  
--return-item-collection-metrics SIZE
```

Conteúdo de `key.json`:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Scared of My Shadow"}  
}
```

Saída:

```
{  
  "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  
    ]  
  }  
}
```

Para obter mais informações, consulte [Gravar um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como excluir um item de forma condicional

O exemplo a seguir exclui um item da tabela ProductCatalog somente se a ProductCategory for Sporting Goods ou Gardening Supplies e o preço estiver entre 500 e 600. Ele retorna detalhes sobre o item que foi excluído.

```
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
```

Conteúdo de names.json:

```
{  
  "#P": "Price"  
}
```

Conteúdo de values.json:

```
{  
  ":cat1": {"S": "Sporting Goods"},  
  ":cat2": {"S": "Gardening Supplies"},  
  ":lo": {"N": "500"},  
  ":hi": {"N": "600"}  
}
```

Saída:

```
{  
  "Attributes": {  
    "Id": {  
      "N": "456"  
    },  
    "Price": {
```



```
        "N": "550"
      },
      "ProductCategory": {
        "S": "Sporting Goods"
      }
    }
  }
}
```

Para obter mais informações, consulte [Gravar um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [Deleteltem](#)na Referência de AWS CLI Comandos.

delete-table

O código de exemplo a seguir mostra como usar delete-table.

AWS CLI

Como excluir uma tabela

O exemplo de delete-table a seguir exclui a tabela MusicCollection.

```
aws dynamodb delete-table \  
  --table-name MusicCollection
```

Saída:

```
{  
  "TableDescription": {  
    "TableStatus": "DELETING",  
    "TableSizeBytes": 0,  
    "ItemCount": 0,  
    "TableName": "MusicCollection",  
    "ProvisionedThroughput": {  
      "NumberOfDecreasesToday": 0,  
      "WriteCapacityUnits": 5,  
      "ReadCapacityUnits": 5  
    }  
  }  
}
```

Para obter mais informações, consulte [Excluir uma tabela](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [DeleteTable](#) na Referência de AWS CLI Comandos.

describe-backup

O código de exemplo a seguir mostra como usar describe-backup.

AWS CLI

Para obter informações sobre um backup existente de uma tabela

O describe-backup exemplo a seguir exibe informações sobre o backup existente especificado.

```
aws dynamodb describe-backup \  
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
  backup/01576616366715-b4e58d3a
```

Saída:

```
{  
  "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": {}
}
}

```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [DescribeBackup](#) na Referência de AWS CLI Comandos.

describe-continuous-backups

O código de exemplo a seguir mostra como usar `describe-continuous-backups`.

AWS CLI

Para obter informações sobre backups contínuos para uma tabela do DynamoDB

O `describe-continuous-backups` exemplo a seguir exibe detalhes sobre as configurações de backup contínuo da `MusicCollection` tabela.

```
aws dynamodb describe-continuous-backups \
  --table-name MusicCollection
```

Saída:

```
{
  "ContinuousBackupsDescription": {
    "ContinuousBackupsStatus": "ENABLED",
```

```
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "DISABLED"  
    }  
  }  
}
```

Para obter mais informações, consulte [Point-in-Time Recovery for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [DescribeContinuousBackups](#) na Referência de AWS CLI Comandos.

describe-contributor-insights

O código de exemplo a seguir mostra como usar `describe-contributor-insights`.

AWS CLI

Para visualizar as configurações do Contributor Insights para uma tabela do DynamoDB

O `describe-contributor-insights` exemplo a seguir exibe as configurações do Contributor Insights para a `MusicCollection` tabela e o índice secundário `AlbumTitle-index` global.

```
aws dynamodb describe-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index
```

Saída:

```
{  
  "TableName": "MusicCollection",  
  "IndexName": "AlbumTitle-index",  
  "ContributorInsightsRuleList": [  
    "DynamoDBContributorInsights-PKC-MusicCollection-1576629651520",  
    "DynamoDBContributorInsights-SKC-MusicCollection-1576629651520",  
    "DynamoDBContributorInsights-PKT-MusicCollection-1576629651520",  
    "DynamoDBContributorInsights-SKT-MusicCollection-1576629651520"  
  ],  
  "ContributorInsightsStatus": "ENABLED",  
  "LastUpdateDateTime": 1576629654.78  
}
```

Para obter mais informações, consulte [Análise do acesso a dados usando o CloudWatch Contributor Insights for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [DescribeContributorInsights](#) na Referência de AWS CLI Comandos.

describe-endpoints

O código de exemplo a seguir mostra como usar `describe-endpoints`.

AWS CLI

Para visualizar informações de endpoints regionais

O `describe-endpoints` exemplo a seguir exibe detalhes sobre os endpoints da AWS região atual.

```
aws dynamodb describe-endpoints
```

Saída:

```
{
  "Endpoints": [
    {
      "Address": "dynamodb.us-west-2.amazonaws.com",
      "CachePeriodInMinutes": 1440
    }
  ]
}
```

Para obter mais informações, consulte [Amazon DynamoDB Endpoints and Quotas](#) na Referência geral.AWS

- Para API obter detalhes, consulte [DescribeEndpoints](#) na Referência de AWS CLI Comandos.

describe-global-table-settings

O código de exemplo a seguir mostra como usar `describe-global-table-settings`.

AWS CLI

Para obter informações sobre as configurações de uma tabela global do DynamoDB

O `describe-global-table-settings` exemplo a seguir exibe as configurações da tabela `MusicCollection` global.

```
aws dynamodb describe-global-table-settings \  
--global-table-name MusicCollection
```

Saída:

```
{  
  "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  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeGlobalTableSettings](#) na Referência de AWS CLI Comandos.

describe-global-table

O código de exemplo a seguir mostra como usar `describe-global-table`.

AWS CLI

Para exibir informações sobre uma tabela global do DynamoDB

O `describe-global-table` exemplo a seguir exibe detalhes sobre a tabela `MusicCollection` global.

```
aws dynamodb describe-global-table \
  --global-table-name MusicCollection
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeGlobalTable](#) na Referência de AWS CLI Comandos.

describe-limits

O código de exemplo a seguir mostra como usar `describe-limits`.

AWS CLI

Para ver os limites da capacidade provisionada

O `describe-limits` exemplo a seguir mostra os limites de capacidade provisionada para sua conta na região atual. AWS

```
aws dynamodb describe-limits
```

Saída:

```
{
  "AccountMaxReadCapacityUnits": 80000,
  "AccountMaxWriteCapacityUnits": 80000,
  "TableMaxReadCapacityUnits": 40000,
  "TableMaxWriteCapacityUnits": 40000
}
```

Para obter mais informações, consulte [Limites no DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [DescribeLimits](#) na Referência de AWS CLI Comandos.

`describe-table-replica-auto-scaling`

O código de exemplo a seguir mostra como usar `describe-table-replica-auto-scaling`.

AWS CLI

Para visualizar as configurações de escalonamento automático em réplicas de uma tabela global

O `describe-table-replica-auto-scaling` exemplo a seguir exibe as configurações de escalonamento automático nas réplicas da tabela `MusicCollection` global.

```
aws dynamodb describe-table-replica-auto-scaling \
  --table-name MusicCollection
```

Saída:

```
{
  "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"
    }
]
}
}

```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [DescribeTableReplicaAutoScaling](#) na Referência de AWS CLI Comandos.

describe-table

O código de exemplo a seguir mostra como usar `describe-table`.

AWS CLI

Como descrever uma tabela

O exemplo a seguir de `describe-table` descreve a tabela `MusicCollection`.

```
aws dynamodb describe-table \  
  --table-name MusicCollection
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Descrever uma tabela](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [DescribeTable](#) na Referência de AWS CLI Comandos.

describe-time-to-live

O código de exemplo a seguir mostra como usar `describe-time-to-live`.

AWS CLI

Como ver as configurações de vida útil de uma tabela

O exemplo `describe-time-to-live` a seguir exibe as configurações de vida útil da tabela `MusicCollection`.

```
aws dynamodb describe-time-to-live \  
  --table-name MusicCollection
```

Saída:

```
{  
  "TimeToLiveDescription": {  
    "TimeToLiveStatus": "ENABLED",  
    "AttributeName": "ttl"  
  }  
}
```

Para obter mais informações, consulte [Vida útil](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [DescribeTimeToLive](#) na Referência de AWS CLI Comandos.

get-item

O código de exemplo a seguir mostra como usar `get-item`.

AWS CLI

Exemplo 1: como ler um item em uma tabela

O exemplo de `get-item` a seguir recupera um item da tabela `MusicCollection`. A tabela tem uma chave `hash-and-range` primária (`ArtisteSongTitle`), portanto, você deve especificar esses dois atributos. O comando também solicita informações sobre a capacidade de leitura consumida pela operação.

```
aws dynamodb get-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-consumed-capacity TOTAL
```

Conteúdo de `key.json`:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Saída:

```
{  
  "Item": {  
    "AlbumTitle": {  
      "S": "Songs About Life"  
    },  
    "SongTitle": {  
      "S": "Happy Day"  
    },  
    "Artist": {  
      "S": "Acme Band"  
    }  
  },  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 0.5  
  }  
}
```

Para obter mais informações, consulte [Ler um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como ler um item usando uma leitura consistente

O exemplo a seguir recupera um item da tabela `MusicCollection` usando leituras altamente consistentes.

```
aws dynamodb get-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --consistent-read \  
  --return-consumed-capacity TOTAL
```

Conteúdo de `key.json`:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Saída:

```
{  
  "Item": {  
    "AlbumTitle": {  
      "S": "Songs About Life"  
    },  
    "SongTitle": {  
      "S": "Happy Day"  
    },  
    "Artist": {  
      "S": "Acme Band"  
    }  
  },  
  "ConsumedCapacity": {  
    "TableName": "MusicCollection",  
    "CapacityUnits": 1.0  
  }  
}
```

Para obter mais informações, consulte [Ler um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 3: como recuperar atributos específicos de um item

O exemplo a seguir usa uma expressão de projeção para recuperar apenas três atributos do item desejado.

```
aws dynamodb get-item \  
  --table-name ProductCatalog \  
  --key '{"Id": {"N": "102"}}' \  
  --projection-expression "#T, #C, #P" \  
  --expression-attribute-names file://names.json
```

Conteúdo de `names.json`:

```
{  
  "#T": "Title",  
  "#C": "ProductCategory",  
  "#P": "Price"  
}
```

Saída:

```
{  
  "Item": {  
    "Price": {  
      "N": "20"  
    },  
    "Title": {  
      "S": "Book 102 Title"  
    },  
    "ProductCategory": {  
      "S": "Book"  
    }  
  }  
}
```

Para obter mais informações, consulte [Ler um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [GetItem](#) na Referência de AWS CLI Comandos.

list-backups

O código de exemplo a seguir mostra como usar `list-backups`.

AWS CLI

Exemplo 1: Para listar todos os backups existentes do DynamoDB

O `list-backups` exemplo a seguir lista todos os seus backups existentes.

```
aws dynamodb list-backups
```

Saída:

```
{
  "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
    }
  ]
}
```



```
}
```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 2: Para listar backups criados pelo usuário em um intervalo de tempo específico

O exemplo a seguir lista somente os backups da MusicCollection tabela que foram criados pelo usuário (não aqueles criados automaticamente pelo DynamoDB) com uma data de criação entre 1º de janeiro de 2020 e 1º de março de 2020.

```
aws dynamodb list-backups \  
  --table-name MusicCollection \  
  --time-range-lower-bound 1577836800 \  
  --time-range-upper-bound 1583020800 \  
  --backup-type USER
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 3: Para limitar o tamanho da página

O exemplo a seguir retorna uma lista de todos os backups existentes, mas recupera somente um item em cada chamada, realizando várias chamadas, se necessário, para obter a lista inteira. Limitar o tamanho da página é útil ao executar os comandos da lista em um grande número de recursos, o que pode resultar em um erro de “tempo limite” ao usar o tamanho de página padrão de 1.000.

```
aws dynamodb list-backups \  
  --page-size 1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 4: Para limitar o número de itens devolvidos

O exemplo a seguir limita o número de itens retornados a 1. A resposta inclui um valor `NextToken` a ser usado para recuperar a próxima página de resultados.

```
aws dynamodb list-backups \  
  --max-items 1
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 5: Para recuperar a próxima página de resultados

O comando a seguir usa o valor `NextToken` de uma chamada anterior feita ao comando `list-backups` para recuperar outra página de resultados. Como a resposta nesse caso não inclui um valor para `NextToken`, sabemos que chegamos ao final dos resultados.

```
aws dynamodb list-backups \
  --starting-
  token abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9
```

Saída

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [ListBackups](#) na Referência de AWS CLI Comandos.

list-contributor-insights

O código de exemplo a seguir mostra como usar `list-contributor-insights`.

AWS CLI

Exemplo 1: Para ver uma lista de resumos do Contributor Insights

O `list-contributor-insights` exemplo a seguir exibe uma lista de resumos do Contributor Insights.

```
aws dynamodb list-contributor-insights
```

Saída:

```
{
  "ContributorInsightsSummaries": [
    {
      "TableName": "MusicCollection",
      "IndexName": "AlbumTitle-index",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "ProductCatalog",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Forum",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Reply",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Thread",
      "ContributorInsightsStatus": "ENABLED"
    }
  ]
}
```

Para obter mais informações, consulte [Análise do acesso a dados usando o CloudWatch Contributor Insights for DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 2: Para limitar o número de itens devolvidos

O exemplo a seguir limita o número de itens devolvidos a 4. A resposta inclui um valor `NextToken` a ser usado para recuperar a próxima página de resultados.

```
aws dynamodb list-contributor-insights \
  --max-results 4
```

Saída:

```
{
```

```

"ContributorInsightsSummaries": [
  {
    "TableName": "MusicCollection",
    "IndexName": "AlbumTitle-index",
    "ContributorInsightsStatus": "ENABLED"
  },
  {
    "TableName": "ProductCatalog",
    "ContributorInsightsStatus": "ENABLED"
  },
  {
    "TableName": "Forum",
    "ContributorInsightsStatus": "ENABLED"
  }
],
"NextToken":
"abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}

```

Para obter mais informações, consulte [Análise do acesso a dados usando o CloudWatch Contributor Insights for DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 3: Para recuperar a próxima página de resultados

O comando a seguir usa o valor `NextToken` de uma chamada anterior feita ao comando `list-contributor-insights` para recuperar outra página de resultados. Como a resposta nesse caso não inclui um valor para `NextToken`, sabemos que chegamos ao final dos resultados.

```

aws dynamodb list-contributor-insights \
  --max-results 4 \
  --next-token abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9

```

Saída:

```

{
  "ContributorInsightsSummaries": [
    {
      "TableName": "Reply",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Thread",

```

```
        "ContributorInsightsStatus": "ENABLED"
      }
    ]
  }
```

Para obter mais informações, consulte [Análise do acesso a dados usando o CloudWatch Contributor Insights for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [ListContributorInsights](#) na Referência de AWS CLI Comandos.

list-global-tables

O código de exemplo a seguir mostra como usar `list-global-tables`.

AWS CLI

Para listar tabelas globais existentes do DynamoDB

O `list-global-tables` exemplo a seguir lista todas as suas tabelas globais existentes.

```
aws dynamodb list-global-tables
```

Saída:

```
{
  "GlobalTables": [
    {
      "GlobalTableName": "MusicCollection",
      "ReplicationGroup": [
        {
          "RegionName": "us-east-2"
        },
        {
          "RegionName": "us-east-1"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [ListGlobalTables](#) na Referência de AWS CLI Comandos.

list-tables

O código de exemplo a seguir mostra como usar `list-tables`.

AWS CLI

Exemplo 1: como listar tabelas

O `list-tables` exemplo a seguir lista todas as tabelas associadas à AWS conta atual e à região.

```
aws dynamodb list-tables
```

Saída:

```
{
  "TableNames": [
    "Forum",
    "ProductCatalog",
    "Reply",
    "Thread"
  ]
}
```

Para obter mais informações, consulte [Listar nomes de tabela](#) no Guia do desenvolvedor Amazon DynamoDB.

Exemplo 2: como limitar o tamanho da página

O exemplo a seguir retorna uma lista de todas as tabelas existentes, mas recupera apenas um item em cada chamada. Pode ser necessário realizar várias chamadas para obter a lista completa. Limitar o tamanho da página é útil ao executar os comandos da lista em um grande número de recursos, o que pode resultar em um erro de “tempo limite” ao usar o tamanho de página padrão de 1.000.

```
aws dynamodb list-tables \
  --page-size 1
```


Saída:

```
{
  "TableNames": [
    "Forum",
    "ProductCatalog",
    "Reply",
    "Thread"
  ]
}
```

Para obter mais informações, consulte [Listar nomes de tabela](#) no Guia do desenvolvedor Amazon DynamoDB.

Exemplo 3: como limitar o número de itens retornados

O exemplo a seguir limita o número de itens retornados para dois. A resposta inclui um valor `NextToken` a ser usado para recuperar a próxima página de resultados.

```
aws dynamodb list-tables \
  --max-items 2
```

Saída:

```
{
  "TableNames": [
    "Forum",
    "ProductCatalog"
  ],
  "NextToken":
  "abCDeFGhiJKlMnOPqrSTuvwXYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Para obter mais informações, consulte [Listar nomes de tabela](#) no Guia do desenvolvedor Amazon DynamoDB.

Exemplo 4: como recuperar a próxima página de resultados

O comando a seguir usa o valor `NextToken` de uma chamada anterior feita ao comando `list-tables` para recuperar outra página de resultados. Como a resposta nesse caso não inclui um valor para `NextToken`, sabemos que chegamos ao final dos resultados.

```
aws dynamodb list-tables \
  --starting-
  token abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Saída:

```
{
  "TableNames": [
    "Reply",
    "Thread"
  ]
}
```

Para obter mais informações, consulte [Listar nomes de tabela](#) no Guia do desenvolvedor Amazon DynamoDB.

- Para API obter detalhes, consulte [ListTables](#) na Referência de AWS CLI Comandos.

list-tags-of-resource

O código de exemplo a seguir mostra como usar `list-tags-of-resource`.

AWS CLI

Exemplo 1: Para listar as tags de um recurso do DynamoDB

O `list-tags-of-resource` exemplo a seguir exibe tags para a `MusicCollection` tabela.

```
aws dynamodb list-tags-of-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

Saída:

```
{
  "Tags": [
    {
      "Key": "Owner",
      "Value": "blueTeam"
    },
    {
      "Key": "Environment",
```

```

        "Value": "Production"
      }
    ]
  }

```

Para obter mais informações, consulte [Tagging for DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 2: Para limitar o número de tags retornadas

O exemplo a seguir limita o número de tags retornadas a 1. A resposta inclui um valor `NextToken` a ser usado para recuperar a próxima página de resultados.

```

aws dynamodb list-tags-of-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
  --max-items 1

```

Saída:

```

{
  "Tags": [
    {
      "Key": "Owner",
      "Value": "blueTeam"
    }
  ],
  "NextToken":
  "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0ppqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}

```

Para obter mais informações, consulte [Tagging for DynamoDB no Amazon DynamoDB Developer Guide](#).

Exemplo 3: Para recuperar a próxima página de resultados

O comando a seguir usa o valor `NextToken` de uma chamada anterior feita ao comando `list-tags-of-resource` para recuperar outra página de resultados. Como a resposta nesse caso não inclui um valor para `NextToken`, sabemos que chegamos ao final dos resultados.

```

aws dynamodb list-tags-of-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \

```

```
--starting-  
token abCDeFGhiJKLmnOPqrSTUvwxyz1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Environment",  
      "Value": "Production"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Tagging for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [ListTagsOfResource](#) na Referência de AWS CLI Comandos.

put-item

O código de exemplo a seguir mostra como usar `put-item`.

AWS CLI

Exemplo 1: como adicionar um item a uma tabela

O `put-item` exemplo a seguir adiciona um novo item à `MusicCollection` tabela.

```
aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item file://item.json \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

Conteúdo de `item.json`:

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Call Me Today"},  
  "AlbumTitle": {"S": "Greatest Hits"}  
}
```

```
}

```

Saída:

```
{
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
  },
  "ItemCollectionMetrics": {
    "ItemCollectionKey": {
      "Artist": {
        "S": "No One You Know"
      }
    },
    "SizeEstimateRangeGB": [
      0.0,
      1.0
    ]
  }
}
```

Para obter mais informações, consulte [Gravar um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como substituir condicionalmente um item em uma tabela

O exemplo de `put-item` a seguir substitui um item existente na tabela `MusicCollection` somente se o item existente tiver um atributo `AlbumTitle` com o valor `Greatest Hits`. O comando retorna o valor anterior do item.

```
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
```

Conteúdo de `item.json`:

```
{

```

```
"Artist": {"S": "No One You Know"},
"SongTitle": {"S": "Call Me Today"},
"AlbumTitle": {"S": "Somewhat Famous"}
}
```

Conteúdo de `names.json`:

```
{
  "#A": "AlbumTitle"
}
```

Conteúdo de `values.json`:

```
{
  ":A": {"S": "Greatest Hits"}
}
```

Saída:

```
{
  "Attributes": {
    "AlbumTitle": {
      "S": "Greatest Hits"
    },
    "Artist": {
      "S": "No One You Know"
    },
    "SongTitle": {
      "S": "Call Me Today"
    }
  }
}
```

Se a chave já existir, você verá a seguinte saída:

```
A client error (ConditionalCheckFailedException) occurred when calling the PutItem
operation: The conditional request failed.
```

Para obter mais informações, consulte [Gravar um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [PutItem](#) na Referência de AWS CLI Comandos.

query

O código de exemplo a seguir mostra como usar query.

AWS CLI

Exemplo 1: como consultar uma tabela

O exemplo da query a seguir consulta itens da tabela `MusicCollection`. A tabela tem uma chave hash-and-range primária (`ArtistSongTitle`), mas essa consulta especifica apenas o valor da chave de hash. Ela retorna nomes de músicas do artista "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
```

Conteúdo de `expression-attributes.json`:

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

Saída:

```
{  
  "Items": [  
    {  
      "SongTitle": {  
        "S": "Call Me Today"  
      },  
      "SongTitle": {  
        "S": "Scared of My Shadow"  
      }  
    }  
  ],  
  "Count": 2,  
}
```

```

    "ScannedCount": 2,
    "ConsumedCapacity": {
      "TableName": "MusicCollection",
      "CapacityUnits": 0.5
    }
  }
}

```

Para obter mais informações, consulte [Operações de consulta no DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como consultar uma tabela usando leituras altamente consistentes e percorrer o índice em ordem decrescente

O exemplo a seguir executa a mesma consulta do primeiro exemplo, mas retorna os resultados na ordem inversa e usa leituras altamente consistentes.

```

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

```

Conteúdo de `expression-attributes.json`:

```

{
  ":v1": {"S": "No One You Know"}
}

```

Saída:

```

{
  "Items": [
    {
      "SongTitle": {
        "S": "Scared of My Shadow"
      }
    },
    {

```



```

        "SongTitle": {
            "S": "Call Me Today"
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}

```

Para obter mais informações, consulte [Operações de consulta no DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 3: como filtrar resultados específicos

O exemplo a seguir consulta o `MusicCollection`, mas exclui os resultados com valores específicos no atributo `AlbumTitle`. Observe que isso não afeta `ScannedCount` ou `ConsumedCapacity` já que o filtro é aplicado após a leitura dos itens.

```

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

```

Conteúdo de `values.json`:

```

{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Blue Sky Blues"},
  ":v3": {"S": "Greatest Hits"}
}

```

Conteúdo de `names.json`:

```

{

```

```
"#n1": "Artist",
"#n2": "AlbumTitle"
}
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Operações de consulta no DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 4: como recuperar somente uma contagem de itens

O exemplo a seguir recupera uma contagem de itens que correspondem à consulta, mas não recupera os itens em si.

```
aws dynamodb query \
  --table-name MusicCollection \
  --select COUNT \
  --key-condition-expression "Artist = :v1" \
  --expression-attribute-values file://expression-attributes.json
```

Conteúdo de `expression-attributes.json`:

```
{
  ":v1": {"S": "No One You Know"}
}
```

Saída:

```
{
  "Count": 2,
  "ScannedCount": 2,
  "ConsumedCapacity": null
}
```

Para obter mais informações, consulte [Operações de consulta no DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 5: como consultar um índice

O exemplo a seguir consulta o índice secundário local `AlbumTitleIndex`. A consulta retorna todos os atributos da tabela base projetados no índice secundário local. Ao consultar um índice secundário local ou global, você deve fornecer o nome da tabela base usando o parâmetro `table-name`.

```
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
```

Conteúdo de `expression-attributes.json`:

```
{
  ":v1": {"S": "No One You Know"}
}
```

Saída:

```
{
  "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
      }
    }
  }
}
```

Para obter mais informações, consulte [Operações de consulta no DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [Consulta](#) na Referência de AWS CLI Comandos.

restore-table-from-backup

O código de exemplo a seguir mostra como usar `restore-table-from-backup`.

AWS CLI

Para restaurar uma tabela do DynamoDB a partir de um backup existente

O `restore-table-from-backup` exemplo a seguir restaura a tabela especificada de um backup existente.

```
aws dynamodb restore-table-from-backup \  
  --target-table-name MusicCollection \  
  --backup-arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
  backup/01576616366715-b4e58d3a
```

Saída:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection2",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ]  
  }  
}
```

```

    ],
    "TableStatus": "CREATING",
    "CreationDateTime": 1576618274.326,
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection2",
    "TableId": "114865c9-5ef3-496c-b4d1-c4cbdd2d44fb",
    "BillingModeSummary": {
      "BillingMode": "PROVISIONED"
    },
    "RestoreSummary": {
      "SourceBackupArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection/backup/01576616366715-b4e58d3a",
      "SourceTableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
MusicCollection",
      "RestoreDateTime": 1576616366.715,
      "RestoreInProgress": true
    }
  }
}

```

Para obter mais informações, consulte [Backup e restauração sob demanda para o DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [RestoreTableFromBackup](#) na Referência de AWS CLI Comandos.

restore-table-to-point-in-time

O código de exemplo a seguir mostra como usar `restore-table-to-point-in-time`.

AWS CLI

Para restaurar uma tabela do DynamoDB em um momento específico

O `restore-table-to-point-in-time` exemplo a seguir restaura a `MusicCollection` tabela no momento especificado.

```
aws dynamodb restore-table-to-point-in-time \  
--source-table-name MusicCollection \  
--target-table-name MusicCollectionRestore \  
--restore-date-time 1576622404.0
```

Saída:

```
{  
  "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
    }
}
}

```

Para obter mais informações, consulte [Point-in-Time Recovery for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [RestoreTableToPointInTime](#) na Referência de AWS CLI Comandos.

scan

O código de exemplo a seguir mostra como usar scan.

AWS CLI

Como verificar uma tabela

O exemplo de scan faz uma varredura da tabela `MusicCollection` e restringe os resultados a músicas do artista “No One You Know”. Em cada item, somente o nome do álbum e da música são retornados.

```

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

```

Conteúdo de `expression-attribute-names.json`:

```

{
  "#ST": "SongTitle",
  "#AT": "AlbumTitle"
}

```



```
}
```

Conteúdo de `expression-attribute-values.json`:

```
{
  ":a": {"S": "No One You Know"}
}
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Trabalhar com verificações no DynamoDB](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [Escanear](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um recurso do DynamoDB

O `tag-resource` exemplo a seguir adiciona um par de chave/valor de tag à `MusicCollection` tabela.

```
aws dynamodb tag-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --tags Key=Owner,Value=blueTeam
```

Este comando não produz saída.

Para obter mais informações, consulte [Tagging for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

`transact-get-items`

O código de exemplo a seguir mostra como usar `transact-get-items`.

AWS CLI

Para recuperar vários itens atomicamente de uma ou mais tabelas

O `transact-get-items` exemplo a seguir recupera vários itens atomicamente.

```
aws dynamodb transact-get-items \  
  --transact-items file://transact-items.json \  
  --return-consumed-capacity TOTAL
```

Conteúdo de `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"
    }
  }
]

```

Saída:

```

{
  "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"
      }
    }
  ]
}

```

Para obter mais informações, consulte [Gerenciamento de fluxos de trabalho complexos com transações do DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [TransactGetItems](#) na Referência de AWS CLI Comandos.

transact-write-items

O código de exemplo a seguir mostra como usar `transact-write-items`.

AWS CLI

Exemplo 1: Para gravar itens atomicamente em uma ou mais tabelas

O `transact-write-items` exemplo a seguir atualiza um item e exclui outro. A operação falhará se uma das operações falhar ou se um dos itens contiver um `Rating` atributo.

```

aws dynamodb transact-write-items \
  --transact-items file://transact-items.json \
  --return-consumed-capacity TOTAL \
  --return-item-collection-metrics SIZE

```

Conteúdo do `transact-items.json` arquivo:

```

[
  {
    "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)"
    }
}
]

```

Saída:

```

{
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 10.0,
            "WriteCapacityUnits": 10.0
        }
    ],
    "ItemCollectionMetrics": {
        "MusicCollection": [
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "No One You Know"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            }
        ],
        {
            "ItemCollectionKey": {

```



```
  },
  {
    "Delete": {
      "Key": {
        "Artist": {"S": "No One You Know"},
        "SongTitle": {"S": "Call Me Today"}
      },
      "TableName": "MusicCollection",
      "ConditionExpression": "attribute_not_exists(Rating)"
    }
  }
]
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de fluxos de trabalho complexos com transações do DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [TransactWriteItems](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso do DynamoDB

O `untag-resource` exemplo a seguir remove a tag com a chave `Owner` da `MusicCollection` tabela.

```
aws dynamodb untag-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
  --tag-keys Owner
```

Este comando não produz saída.

Para obter mais informações, consulte [Tagging for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-continuous-backups

O código de exemplo a seguir mostra como usar `update-continuous-backups`.

AWS CLI

Para atualizar as configurações de backup contínuo para uma tabela do DynamoDB

O `update-continuous-backups` exemplo a seguir permite a point-in-time recuperação da `MusicCollection` tabela.

```
aws dynamodb update-continuous-backups \  
  --table-name MusicCollection \  
  --point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

Saída:

```
{  
  "ContinuousBackupsDescription": {  
    "ContinuousBackupsStatus": "ENABLED",  
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "ENABLED",  
      "EarliestRestorableDateTime": 1576622404.0,  
      "LatestRestorableDateTime": 1576622404.0  
    }  
  }  
}
```

Para obter mais informações, consulte [Point-in-Time Recovery for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [UpdateContinuousBackups](#) na Referência de AWS CLI Comandos.

update-contributor-insights

O código de exemplo a seguir mostra como usar `update-contributor-insights`.

AWS CLI

Para habilitar o Contributor Insights em uma tabela

O `update-contributor-insights` exemplo a seguir ativa o Contributor Insights na `MusicCollection` tabela e no índice secundário `AlbumTitle-index` global.

```
aws dynamodb update-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index \  
  --contributor-insights-action ENABLE
```

Saída:

```
{  
  "TableName": "MusicCollection",  
  "IndexName": "AlbumTitle-index",  
  "ContributorInsightsStatus": "ENABLING"  
}
```

Para obter mais informações, consulte [Análise do acesso a dados usando o CloudWatch Contributor Insights for DynamoDB no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [UpdateContributorInsights](#) na Referência de AWS CLI Comandos.

update-global-table-settings

O código de exemplo a seguir mostra como usar `update-global-table-settings`.

AWS CLI

Para atualizar as configurações de capacidade de gravação provisionada em uma tabela global do DynamoDB

O `update-global-table-settings` exemplo a seguir define a capacidade de gravação provisionada da tabela `MusicCollection` global como 15.

```
aws dynamodb update-global-table-settings \  
  --global-table-name MusicCollection \  
  --global-table-provisioned-write-capacity-units 15
```

Saída:

```
{
```

```
"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
    }
  }
]
}
```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [UpdateGlobalTableSettings](#) na Referência de AWS CLI Comandos.

update-global-table

O código de exemplo a seguir mostra como usar `update-global-table`.

AWS CLI

Para atualizar uma tabela global do DynamoDB

O `update-global-table` exemplo a seguir adiciona uma réplica na região especificada à tabela `MusicCollection` global.

```
aws dynamodb update-global-table \  
  --global-table-name MusicCollection \  
  --replica-updates Create={RegionName=eu-west-1}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [UpdateGlobalTable](#) na Referência de AWS CLI Comandos.

update-item

O código de exemplo a seguir mostra como usar `update-item`.

AWS CLI

Exemplo 1: como atualizar um item em uma tabela

O exemplo da `update-item` a seguir atualiza um item da tabela `MusicCollection`. Ele adiciona um novo atributo (`Year`) e modifica o atributo `AlbumTitle`. Todos os atributos no item, conforme aparecem após a atualização, são retornados na resposta.

```
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
```

Conteúdo de `key.json`:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Conteúdo de `expression-attribute-names.json`:

```
{  
  "#Y": "Year", "#AT": "AlbumTitle"  
}
```

Conteúdo de `expression-attribute-values.json`:

```
{
  "y":{"N": "2015"},
  "t":{"S": "Louder Than Ever"}
}
```

Saída:

```
{
  "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
  ]
}
```

Para obter mais informações, consulte [Gravar um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como atualizar um item de forma condicional

O exemplo a seguir atualiza um item na tabela `MusicCollection`, mas somente se o item existente ainda não tiver um atributo `Year`.

```
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)"
```

Conteúdo de `key.json`:

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Conteúdo de `expression-attribute-names.json`:

```
{  
  "#Y": "Year",  
  "#AT": "AlbumTitle"  
}
```

Conteúdo de `expression-attribute-values.json`:

```
{  
  ":y": {"N": "2015"},  
  ":t": {"S": "Louder Than Ever"}  
}
```

Se o item já tiver um atributo `Year`, o DynamoDB retornará saída a seguir.

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem  
operation: The conditional request failed
```

Para obter mais informações, consulte [Gravar um item](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [UpdateItem](#) na Referência de AWS CLI Comandos.

update-table-replica-auto-scaling

O código de exemplo a seguir mostra como usar `update-table-replica-auto-scaling`.

AWS CLI

Para atualizar as configurações de escalonamento automático em réplicas de uma tabela global

O `update-table-replica-auto-scaling` exemplo a seguir atualiza as configurações de escalonamento automático da capacidade de gravação em réplicas da tabela global especificada.

```
aws dynamodb update-table-replica-auto-scaling \  
  --table-name MusicCollection \  
  --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

Conteúdo de `auto-scaling-policy.json`:

```
{  
  "MinimumUnits": 10,  
  "MaximumUnits": 100,  
  "AutoScalingDisabled": false,  
  "ScalingPolicyUpdate": {  
    "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",  
    "TargetTrackingScalingPolicyConfiguration": {  
      "TargetValue": 80  
    }  
  }  
}
```

Saída:

```
{  
  "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"
}
]
}
}

```

Para obter mais informações, consulte Tabelas [globais do DynamoDB](#) no Amazon DynamoDB Developer Guide.

- Para API obter detalhes, consulte [UpdateTableReplicaAutoScaling](#) na Referência de AWS CLI Comandos.

update-table

O código de exemplo a seguir mostra como usar `update-table`.

AWS CLI

Exemplo 1: como modificar o modo de faturamento de uma tabela

O exemplo `update-table` a seguir aumenta a capacidade de leitura e gravação provisionada na tabela `MusicCollection`.

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --billing-mode PROVISIONED \  
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

```

        "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"
    }
}
}
}

```

Para ter mais informações, consulte [Updating a Table](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 2: como criar um índice secundário global

O exemplo a seguir adiciona um índice secundário global à tabela `MusicCollection`.

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \
  --global-secondary-index-updates file://gsi-updates.json

```

Conteúdo de `gsi-updates.json`:

```

[
  {
    "Create": {
      "IndexName": "AlbumTitle-index",
      "KeySchema": [
        {
          "AttributeName": "AlbumTitle",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": 10,
        "WriteCapacityUnits": 10
      },
      "Projection": {
        "ProjectionType": "ALL"
      }
    }
  }
]

```

```

    }
  }
]

```

Saída:

```

{
  "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"
      }
    ]
  }
}

```

Para ter mais informações, consulte [Updating a Table](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 3: como habilitar o DynamoDB Streams em uma tabela

O comando a seguir habilita o DynamoDB Streams na tabela MusicCollection.

```
aws dynamodb update-table \
```

```
--table-name MusicCollection \  
--stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE
```

Saída:

```
{  
  "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"
}
}

```

Para ter mais informações, consulte [Updating a Table](#) no Guia do desenvolvedor do Amazon DynamoDB.

Exemplo 4: como habilitar a criptografia do lado do servidor

O exemplo a seguir habilita a criptografia do lado do servidor na tabela `MusicCollection`.

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --sse-specification Enabled=true,SSEType=KMS

```

Saída:

```

{
  "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"
    }
  }
}
```

Para ter mais informações, consulte [Updating a Table](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [UpdateTable](#) na Referência de AWS CLI Comandos.

update-time-to-live

O código de exemplo a seguir mostra como usar `update-time-to-live`.

AWS CLI

Como atualizar as configurações de vida útil de uma tabela

O exemplo `update-time-to-live` a seguir habilita a vida útil na tabela especificada.

```
aws dynamodb update-time-to-live \
  --table-name MusicCollection \
  --time-to-live-specification Enabled=true,AttributeName=ttl
```

Saída:

```
{
  "TimeToLiveSpecification": {
    "Enabled": true,
    "AttributeName": "ttl"
  }
}
```

Para obter mais informações, consulte [Vida útil](#) no Guia do desenvolvedor do Amazon DynamoDB.

- Para API obter detalhes, consulte [UpdateTimeToLive](#) na Referência de AWS CLI Comandos.

Exemplos do DynamoDB Streams usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o DynamoDB Streams.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-stream

O código de exemplo a seguir mostra como usar `describe-stream`.

AWS CLI

Para obter informações sobre um stream do DynamoDB

O `describe-stream` comando a seguir exibe informações sobre o stream específico do DynamoDB.

```
aws dynamodbstreams describe-stream \  
  --stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/  
stream/2019-10-22T18:02:01.576
```

Saída:

```
{  
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Capturando a atividade da tabela com o DynamoDB Streams no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [DescribeStream](#) na Referência de AWS CLI Comandos.

get-records

O código de exemplo a seguir mostra como usar `get-records`.

AWS CLI

Para obter registros de um stream do Dynamodb

O `get-records` comando a seguir recupera registros usando o iterador de fragmento especificado do Amazon DynamoDB.

```
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
+hFxFAWR5C7QIL0XPc8mRBfNIazfrVCjJK8/jsjCzsqMyXKzJbhh+GXcoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPI dmTRG
+w/LEGS05ha1qNP+VL4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+DtcM5BHayBs
+PVYHITaTLiInFLT
+LCwvaz1QH3MY3b8A05Z800wjpkM60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw=="
```

Saída:

```
{
  "Records": [
    {
      "eventID": "c3b5d798eef6215d42f8137b19a88e50",
      "eventName": "INSERT",
      "eventVersion": "1.1",
      "eventSource": "aws:dynamodb",
      "awsRegion": "us-west-1",
      "dynamodb": {
        "ApproximateCreationDateTime": 1571849028.0,
        "Keys": {
          "Artist": {
            "S": "No One You Know"
          },
          "SongTitle": {
            "S": "Call Me Today"
          }
        },
        "NewImage": {
          "AlbumTitle": {
            "S": "Somewhat Famous"
          },
          "Artist": {
            "S": "No One You Know"
          },
          "Awards": {
            "N": "1"
          }
        }
      }
    }
  ]
}
```

```
        },
        "SongTitle": {
            "S": "Call Me Today"
        }
    },
    "SequenceNumber": "700000000013256296913",
    "SizeBytes": 119,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
}
},
{
    "eventID": "878960a6967867e2da16b27380a27328",
    "eventName": "INSERT",
    "eventVersion": "1.1",
    "eventSource": "aws:dynamodb",
    "awsRegion": "us-west-1",
    "dynamodb": {
        "ApproximateCreationDateTime": 1571849029.0,
        "Keys": {
            "Artist": {
                "S": "Acme Band"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        },
        "NewImage": {
            "AlbumTitle": {
                "S": "Songs About Life"
            },
            "Artist": {
                "S": "Acme Band"
            },
            "Awards": {
                "N": "10"
            },
            "SongTitle": {
                "S": "Happy Day"
            }
        },
        "SequenceNumber": "800000000013256297217",
        "SizeBytes": 100,
        "StreamViewType": "NEW_AND_OLD_IMAGES"
    }
}
```



```
},
{
  "eventID": "520fabde080e159fc3710b15ee1d4daa",
  "eventName": "MODIFY",
  "eventVersion": "1.1",
  "eventSource": "aws:dynamodb",
  "awsRegion": "us-west-1",
  "dynamodb": {
    "ApproximateCreationDateTime": 1571849734.0,
    "Keys": {
      "Artist": {
        "S": "Acme Band"
      },
      "SongTitle": {
        "S": "Happy Day"
      }
    },
    "NewImage": {
      "AlbumTitle": {
        "S": "Updated Album Title"
      },
      "Artist": {
        "S": "Acme Band"
      },
      "Awards": {
        "N": "10"
      },
      "SongTitle": {
        "S": "Happy Day"
      }
    },
    "OldImage": {
      "AlbumTitle": {
        "S": "Songs About Life"
      },
      "Artist": {
        "S": "Acme Band"
      },
      "Awards": {
        "N": "10"
      },
      "SongTitle": {
        "S": "Happy Day"
      }
    }
  }
}
```

```

    },
    "SequenceNumber": "900000000013256687845",
    "SizeBytes": 170,
    "StreamViewType": "NEW_AND_OLD_IMAGES"
  }
}
],
  "NextShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/
Music/stream/2019-10-23T16:41:08.740|1|AAAAAAAAAAAEhEI04jkFLW
+LK0wivjT8d/IHEh3iExV2xK00aTxEzVy1C1C7Kbb5+Z0W6bT9VQ2n1/
mrs7+PRia0ZCHJu7JHJVW7zlsq0i/ges3fw8GYEymyL+piEk35cx67rQqwKKyq
+Q6w9JyjreIOj4F2lWLV26lBwRTrIYC4IB7C3BZZK4715QwYdDxNdVHiSBRZX8UqoS6W0t0F87xZLNB9F/
NhYBLXi/wcGvAcBcC0TNI0H+N0Nqwt0B/
FGckNrf8YZ0xRoNN6RgGuVWHF3px0hxEJeFZoSoJTIKeG9YcYxzi5Ci/
mhdTm7tBXnbw5c6xmsGsBqTirNjldyJLcWl8Cl0UOLX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0fF/
ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWf2wvWt6Iy2xGrR2r2HzYps9dwG0arVdEITaJfWzNoL4HajMhmREZ
+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PvV6iknY6d19MrdbSTMkoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da
}

```

Para obter mais informações, consulte [Capturando a atividade da tabela com o DynamoDB Streams no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [GetRecords](#) na Referência de AWS CLI Comandos.

get-shard-iterator

O código de exemplo a seguir mostra como usar `get-shard-iterator`.

AWS CLI

Para obter um iterador de fragmentos

O `get-shard-iterator` comando a seguir recupera um iterador de fragmento para o fragmento especificado.

```

aws dynamodbstreams get-shard-iterator \
  --stream-arn arn:aws:dynamodb:us-west-1:12356789012:table/Music/  

stream/2019-10-22T18:02:01.576 \
  --shard-id shardId-00000001571780995058-40810d86 \
  --shard-iterator-type LATEST

```

Saída:

```
{
  "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/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaxNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/1EGS05ha1qNP+V14+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+DtcM5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHndusw==
}
```

Para obter mais informações, consulte [Capturando a atividade da tabela com o DynamoDB Streams no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [GetShardIterator](#) na Referência de AWS CLI Comandos.

list-streams

O código de exemplo a seguir mostra como usar `list-streams`.

AWS CLI

Para listar streams do DynamoDB

O `list-streams` comando a seguir lista todos os streams existentes do Amazon DynamoDB dentro da região padrão. AWS

```
aws dynamodbstreams list-streams
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Capturando a atividade da tabela com o DynamoDB Streams no Amazon DynamoDB Developer Guide](#).

- Para API obter detalhes, consulte [ListStreams](#) na Referência de AWS CLI Comandos.

EC2Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonEC2.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-address-transfer

O código de exemplo a seguir mostra como usar `accept-address-transfer`.

AWS CLI

Para aceitar um endereço IP elástico transferido para sua conta

O `accept-address-transfer` exemplo a seguir aceita a transferência do endereço IP elástico especificado para sua conta.

```
aws ec2 accept-address-transfer \  
  --address 100.21.184.216
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Transferir endereços IP elásticos](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [AcceptAddressTransfer](#) na Referência de AWS CLI Comandos.

accept-reserved-instances-exchange-quote

O código de exemplo a seguir mostra como usar `accept-reserved-instances-exchange-quote`.

AWS CLI

Para realizar uma troca de instância reservada conversível

Este exemplo executa uma troca das instâncias reservadas conversíveis especificadas.

Comando:

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

Saída:

```
{
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"
}
```

- Para API obter detalhes, consulte [AcceptReservedInstancesExchangeQuote](#) na Referência de AWS CLI Comandos.

accept-transit-gateway-peering-attachment

O código de exemplo a seguir mostra como usar `accept-transit-gateway-peering-attachment`.

AWS CLI

Para aceitar um anexo de emparelhamento do Transit Gateway

O `accept-transit-gateway-peering-attachment` exemplo a seguir aceita o anexo de emparelhamento do gateway de trânsito especificado. O `--region` parâmetro especifica a região na qual o gateway de trânsito aceitante está localizado.

```
aws ec2 accept-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
  --region us-east-2
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Transit Gateway Peering Attachments](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [AcceptTransitGatewayPeeringAttachment](#) na Referência de AWS CLI Comandos.

accept-transit-gateway-vpc-attachment

O código de exemplo a seguir mostra como usar `accept-transit-gateway-vpc-attachment`.

AWS CLI

Aceitar uma solicitação para anexar um VPC a um gateway de trânsito.

O `accept-transit-gateway-vpc-attachment` exemplo a seguir aceita a solicitação para um anexo especificado.

```
aws ec2 accept-transit-gateway-vpc-attachment \  
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Transit Gateway Attachments to a VPC](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [AcceptTransitGatewayVpcAttachment](#) na Referência de AWS CLI Comandos.

accept-vpc-endpoint-connections

O código de exemplo a seguir mostra como usar `accept-vpc-endpoint-connections`.

AWS CLI

Para aceitar uma solicitação de conexão de endpoint de interface

Este exemplo aceita a solicitação de conexão de endpoint especificada para o serviço de endpoint especificado.

Comando:

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --  
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Saída:

```
{  
  "Unsuccessful": []  
}
```

- Para API obter detalhes, consulte [AcceptVpcEndpointConnections](#) na Referência de AWS CLI Comandos.

accept-vpc-peering-connection

O código de exemplo a seguir mostra como usar `accept-vpc-peering-connection`.

AWS CLI

Para aceitar uma conexão de VPC emparelhamento

Este exemplo aceita a solicitação de conexão de VPC emparelhamento especificada.

Comando:

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Saída:

```
{
```



```

"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"
  }
}
}
}

```

- Para API obter detalhes, consulte [AcceptVpcPeeringConnection](#) na Referência de AWS CLI Comandos.

advertise-byoip-cidr

O código de exemplo a seguir mostra como usar `advertise-byoip-cidr`.

AWS CLI

Para anunciar um intervalo de endereços

O `advertise-byoip-cidr` exemplo a seguir anuncia o intervalo de IPv4 endereços públicos especificado.

```

aws ec2 advertise-byoip-cidr \
  --cidr 203.0.113.25/24

```

Saída:

```

{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",

```

```
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    "State": "provisioned"
  }
}
```

- Para API obter detalhes, consulte [AdvertiseByoipCidr](#) na Referência de AWS CLI Comandos.

allocate-address

O código de exemplo a seguir mostra como usar `allocate-address`.

AWS CLI

Exemplo 1: alocar um endereço IP elástico do conjunto de endereços da Amazon

O exemplo `allocate-address` a seguir aloca um endereço IP elástico. A Amazon EC2 seleciona o endereço do pool de endereços da Amazon.

```
aws ec2 allocate-address
```

Saída:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-01435ba59eEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2",
  "Domain": "vpc"
}
```

Para obter mais informações, consulte [Endereços IP elásticos](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: alocar um endereço IP elástico e associá-lo a um grupo de borda de rede

O exemplo `allocate-address` a seguir aloca um endereço IP elástico e o associa ao grupo de borda de rede especificado.

```
aws ec2 allocate-address \
  --network-border-group us-west-2-lax-1
```

Saída:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2-lax-1",
  "Domain": "vpc"
}
```

Para obter mais informações, consulte [Endereços IP elásticos](#) no Guia EC2 do usuário da Amazon.

Exemplo 3: alocar um endereço IP elástico de um conjunto de endereços de sua propriedade

O exemplo `allocate-address` a seguir aloca um endereço IP elástico de um conjunto de endereços que você trouxe para a conta da Amazon Web Services. A Amazon EC2 seleciona o endereço do pool de endereços.

```
aws ec2 allocate-address \
  --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

Saída:

```
{
  "AllocationId": "eipalloc-02463d08ceEXAMPLE",
  "NetworkBorderGroup": "us-west-2",
  "CustomerOwnedIp": "18.218.95.81",
  "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",
  "Domain": "vpc",
  "NetworkBorderGroup": "us-west-2",
}
```

Para obter mais informações, consulte [Endereços IP elásticos](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [AllocateAddress](#) na Referência de AWS CLI Comandos.

allocate-hosts

O código de exemplo a seguir mostra como usar `allocate-hosts`.

AWS CLI

Exemplo 1: Para alocar um host dedicado

O `allocate-hosts` exemplo a seguir aloca um único host dedicado na zona de `eu-west-1a` disponibilidade, no qual você pode executar `m5.large` instâncias. Por padrão, o host dedicado aceita somente a execução da instância de destino e não oferece suporte à recuperação do host.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --quantity 1
```

Saída:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Exemplo 2: Para alocar um host dedicado com posicionamento automático e recuperação de host ativados

O `allocate-hosts` exemplo a seguir aloca um único host dedicado na zona de `eu-west-1a` disponibilidade com posicionamento automático e recuperação de host ativados.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --auto-placement on \  
  --host-recovery on \  
  --quantity 1
```

Saída:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

```
}
```

Exemplo 3: Para alocar um host dedicado com tags

O `allocate-hosts` exemplo a seguir aloca um único host dedicado e aplica uma tag com uma chave chamada `purpose` e um valor de `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}'
```

Saída:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Para obter mais informações, consulte [Alocação de hosts dedicados](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [AllocateHosts](#) na Referência de AWS CLI Comandos.

`allocate-ipam-pool-cidr`

O código de exemplo a seguir mostra como usar `allocate-ipam-pool-cidr`.

AWS CLI

Para alocar um CIDR de um pool IPAM

O `allocate-ipam-pool-cidr` exemplo a seguir aloca a CIDR de um IPAM pool.

(Linux):

```
aws ec2 allocate-ipam-pool-cidr \  
  --ipam-pool-id ipam-pool-0533048da7d823723 \  
  --cidr 10.0.0.0/24
```

```
--netmask-length 24
```

(Windows):

```
aws ec2 allocate-ipam-pool-cidr ^  
--ipam-pool-id ipam-pool-0533048da7d823723 ^  
--netmask-length 24
```

Saída:

```
{  
  "IpamPoolAllocation": {  
    "Cidr": "10.0.0.0/24",  
    "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfb",  
    "ResourceType": "custom",  
    "ResourceOwner": "123456789012"  
  }  
}
```

Para obter mais informações, consulte [Alocar manualmente um CIDR a um pool para reservar espaço de endereço IP](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [AllocateIpamPoolCidr](#) na Referência de AWS CLI Comandos.

apply-security-groups-to-client-vpn-target-network

O código de exemplo a seguir mostra como usar `apply-security-groups-to-client-vpn-target-network`.

AWS CLI

Para aplicar grupos de segurança a uma rede de destino para um VPN endpoint do cliente

O `apply-security-groups-to-client-vpn-target-network` exemplo a seguir aplica o grupo de segurança `sg-01f6e627a89f4db32` à associação entre a rede de destino especificada e o VPN endpoint do cliente.

```
aws ec2 apply-security-groups-to-client-vpn-target-network \  
--security-group-ids sg-01f6e627a89f4db32 \  
--vpc-id vpc-0e2110c2f324332e0 \  
--netmask-length 24
```

```
--client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Saída:

```
{
  "SecurityGroupIds": [
    "sg-01f6e627a89f4db32"
  ]
}
```

Para obter mais informações, consulte [Target Networks](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [ApplySecurityGroupsToClientVpnTargetNetwork](#) na Referência de AWS CLI Comandos.

assign-ipv6-addresses

O código de exemplo a seguir mostra como usar `assign-ipv6-addresses`.

AWS CLI

Para atribuir IPv6 endereços específicos a uma interface de rede

Este exemplo atribui os IPv6 endereços especificados à interface de rede especificada.

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
```

Saída:

```
{
  "AssignedIpv6Addresses": [
    "2001:db8:1234:1a00:3304:8879:34cf:4071",
    "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

Para atribuir IPv6 endereços que a Amazon seleciona a uma interface de rede

Este exemplo atribui dois IPv6 endereços à interface de rede especificada. A Amazon atribui automaticamente esses IPv6 endereços a partir dos IPv6 endereços disponíveis no intervalo de IPv6 CIDR blocos da sub-rede.

Comando:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-count 2
```

Saída:

```
{
  "AssignedIpv6Addresses": [
    "2001:db8:1234:1a00:3304:8879:34cf:4071",
    "2001:db8:1234:1a00:9691:9503:25ad:1761"
  ],
  "NetworkInterfaceId": "eni-38664473"
}
```

- Para API obter detalhes, consulte [AssignIpv6Addresses](#) in AWS CLI Command Reference.

assign-private-ip-addresses

O código de exemplo a seguir mostra como usar `assign-private-ip-addresses`.

AWS CLI

Para atribuir um endereço IP privado secundário específico a uma interface de rede

Este exemplo atribui o endereço IP privado secundário especificado à interface de rede especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

Para atribuir endereços IP privados secundários que a Amazon EC2 seleciona a uma interface de rede

Este exemplo atribui dois endereços IP privados secundários à interface de rede especificada. A Amazon atribui EC2 automaticamente esses endereços IP a partir dos endereços IP disponíveis no intervalo de CIDR blocos da sub-rede à qual a interface de rede está associada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-private-ip-address-count 2
```

- Para API obter detalhes, consulte [AssignPrivateIpAddresses](#) na Referência de AWS CLI Comandos.

assign-private-nat-gateway-address

O código de exemplo a seguir mostra como usar `assign-private-nat-gateway-address`.

AWS CLI

Para atribuir endereços IP privados ao seu NAT gateway privado

O `assign-private-nat-gateway-address` exemplo a seguir atribui dois endereços IP privados ao NAT gateway privado especificado.

```
aws ec2 assign-private-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --private-ip-address-count 2
```

Saída:

```
{  
  "NatGatewayId": "nat-1234567890abcdef0",  
  "NatGatewayAddresses": [  
    {  
      "NetworkInterfaceId": "eni-0065a61b324d1897a",  
      "IsPrimary": false,  
      "Status": "assigning"  
    },  
    {  
      "NetworkInterfaceId": "eni-0065a61b324d1897a",  
      "IsPrimary": false,
```

```
        "Status": "assigning"
      }
    ]
  }
```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [AssignPrivateNatGatewayAddress](#) na Referência de AWS CLI Comandos.

associate-address

O código de exemplo a seguir mostra como usar `associate-address`.

AWS CLI

Para associar endereços IP elásticos em EC2 -Classic

Este exemplo associa um endereço IP elástico a uma instância em EC2 -Classic. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 associate-address --instance-id i-07ffe74c7330ebf53 --public-ip 198.51.100.0
```

Para associar um endereço IP elástico em EC2 - VPC

Este exemplo associa um endereço IP elástico a uma instância em um VPC.

Comando:

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-id eipalloc-64d5890a
```

Saída:

```
{
  "AssociationId": "eipassoc-2bebb745"
}
```

Este exemplo associa um endereço IP elástico a uma interface de rede.

Comando:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d
```

Este exemplo associa um IP elástico a um endereço IP privado associado a uma interface de rede.

Comando:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d --private-ip-address 10.0.0.85
```

- Para API obter detalhes, consulte [AssociateAddress](#) na Referência de AWS CLI Comandos.

associate-client-vpn-target-network

O código de exemplo a seguir mostra como usar `associate-client-vpn-target-network`.

AWS CLI

Para associar uma rede de destino a um VPN endpoint do cliente

O `associate-client-vpn-target-network` exemplo a seguir associa uma sub-rede ao endpoint do cliente VPN especificado.

```
aws ec2 associate-client-vpn-target-network \  
  --subnet-id subnet-0123456789abcabca \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Saída:

```
{  
  "AssociationId": "cvpn-assoc-12312312312312312",  
  "Status": {  
    "Code": "associating"  
  }  
}
```

Para obter mais informações, consulte [Target Networks](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [AssociateClientVpnTargetNetwork](#) na Referência de AWS CLI Comandos.

associate-dhcp-options

O código de exemplo a seguir mostra como usar `associate-dhcp-options`.

AWS CLI

Para associar um conjunto de DHCP opções ao seu VPC

Este exemplo associa o conjunto DHCP de opções especificado ao especificado VPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

Para associar as DHCP opções padrão definidas ao seu VPC

Este exemplo associa o conjunto DHCP de opções padrão ao especificado VPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- Para API obter detalhes, consulte [AssociateDhcpOptions](#) na Referência de AWS CLI Comandos.

associate-iam-instance-profile

O código de exemplo a seguir mostra como usar `associate-iam-instance-profile`.

AWS CLI

Para associar um perfil de IAM instância a uma instância

Este exemplo associa um perfil de IAM instância chamado `admin-role` com `instancei-123456789abcde123`.

Comando:

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-  
instance-profile Name=admin-role
```

Saída:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-123456789abcde123",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0e7736511a163c209",  
    "IamInstanceProfile": {  
      "Id": "AIPAJBLK7RKJKWDXVHIEC",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [AssociatelaInstanceProfile](#) na Referência de AWS CLI Comandos.

associate-instance-event-window

O código de exemplo a seguir mostra como usar `associate-instance-event-window`.

AWS CLI

Exemplo 1: Para associar uma ou mais instâncias a uma janela de eventos

O `associate-instance-event-window` exemplo a seguir associa uma ou mais instâncias a uma janela de eventos.

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Saída:

```
{
```

```

    "InstanceEventWindow": {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "AssociationTarget": {
        "InstanceIds": [
          "i-1234567890abcdef0",
          "i-0598c7d356eba48d7"
        ],
        "Tags": [],
        "DedicatedHostIds": []
      },
      "State": "creating"
    }
  }
}

```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 2: Para associar tags de instância a uma janela de eventos

O `associate-instance-event-window` exemplo a seguir associa tags de instância a uma janela de eventos. Insira um `instance-event-window-id` parâmetro para especificar a janela do evento. Para associar tags de instância, especifique o `association-target` parâmetro e, para o valor do parâmetro, especifique uma ou mais tags.

```

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}]"

```

Saída:

```

{
  "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"
}

```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 3: Para associar um host dedicado a uma janela de eventos

O `associate-instance-event-window` exemplo a seguir associa um host dedicado a uma janela de eventos. Insira um `instance-event-window-id` parâmetro para especificar a janela do evento. Para associar um host dedicado, especifique o `--association-target` parâmetro e, para os valores dos parâmetros, especifique um ou mais hosts dedicados IDs.

```

aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "DedicatedHostIds=h-029fa35a02b99801d"

```

Saída:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": [
        "h-029fa35a02b99801d"
      ]
    }
  }
}

```

```
    },  
    "State": "creating"  
  }  
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [AssociateInstanceEventWindow](#) na Referência de AWS CLI Comandos.

associate-ipam-resource-discovery

O código de exemplo a seguir mostra como usar `associate-ipam-resource-discovery`.

AWS CLI

Para associar uma descoberta de recursos a um IPAM

Neste exemplo, você é um administrador IPAM delegado e uma descoberta de recursos foi criada e compartilhada com você por outra AWS conta para que você possa usá-la IPAM para gerenciar e monitorar recursos CIDRs pertencentes à outra conta.

Observação

Para concluir essa solicitação, você precisará da ID de descoberta de recursos que você pode obter com [describe-ipam-resource-discoveries](#) da IPAM ID que você pode obter com [describe-ipams](#). A descoberta do recurso que você está associando deve ter sido compartilhada primeiro com sua conta usando AWS RAM .A que `--region` você digitou deve corresponder à região de origem da que você está associando. IPAM

O `associate-ipam-resource-discovery` exemplo a seguir associa uma descoberta de recursos a um 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
```


Saída:

```
{
  {
    "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": []
    }
  }
}
```

Depois de associar uma descoberta de recursos, você pode monitorar e/ou gerenciar os endereços IP dos recursos criados pelas outras contas. Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [AssociateIpamResourceDiscovery](#) na Referência de AWS CLI Comandos.

associate-nat-gateway-address

O código de exemplo a seguir mostra como usar `associate-nat-gateway-address`.

AWS CLI

Para associar um endereço IP elástico a um NAT gateway público

O `associate-nat-gateway-address` exemplo a seguir associa o endereço IP elástico especificado ao NAT gateway público especificado. AWS atribui automaticamente um IPv4 endereço privado secundário.

```
aws ec2 associate-nat-gateway-address \
```

```
--nat-gateway-id nat-1234567890abcdef0 \  
--allocation-ids eipalloc-0be6ecac95EXAMPLE
```

Saída:

```
{  
  "NatGatewayId": "nat-1234567890abcdef0",  
  "NatGatewayAddresses": [  
    {  
      "AllocationId": "eipalloc-0be6ecac95EXAMPLE",  
      "NetworkInterfaceId": "eni-09cc4b2558794f7f9",  
      "IsPrimary": false,  
      "Status": "associating"  
    }  
  ]  
}
```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [AssociateNatGatewayAddress](#) na Referência de AWS CLI Comandos.

associate-route-table

O código de exemplo a seguir mostra como usar `associate-route-table`.

AWS CLI

Para associar uma tabela de rotas a uma sub-rede

Este exemplo associa a tabela de rotas especificada à sub-rede especificada.

Comando:

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-  
id subnet-9d4a7b6c
```

Saída:

```
{  
  "AssociationId": "rtbassoc-781d0d1a"  
}
```

- Para API obter detalhes, consulte [AssociateRouteTable](#) na Referência de AWS CLI Comandos.

associate-subnet-cidr-block

O código de exemplo a seguir mostra como usar `associate-subnet-cidr-block`.

AWS CLI

Para associar um IPv6 CIDR bloco a uma sub-rede

Este exemplo associa um IPv6 CIDR bloco à sub-rede especificada.

Comando:

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-block 2001:db8:1234:1a00::/64
```

Saída:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
    "AssociationId": "subnet-cidr-assoc-3aa54053",
    "Ipv6CidrBlockState": {
      "State": "associating"
    }
  }
}
```

- Para API obter detalhes, consulte [AssociateSubnetCidrBlock](#) na Referência de AWS CLI Comandos.

associate-transit-gateway-multicast-domain

O código de exemplo a seguir mostra como usar `associate-transit-gateway-multicast-domain`.

AWS CLI

Para associar um gateway de trânsito a um domínio multicast

O `associate-transit-gateway-multicast-domain` exemplo a seguir associa a sub-rede e o anexo especificados ao domínio multicast especificado.

```
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
```

Saída:

```
{
  "Associations": [
    {
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
      "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",
      "ResourceId": "vpc-01128d2c240c09bd5",
      "ResourceType": "vpc",
      "Subnets": [
        {
          "SubnetId": "subnet-000de86e3b49c932a",
          "State": "associating"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [AssociateTransitGatewayMulticastDomain](#) na Referência de AWS CLI Comandos.

associate-transit-gateway-route-table

O código de exemplo a seguir mostra como usar `associate-transit-gateway-route-table`.

AWS CLI

Para associar uma tabela de rotas do Transit Gateway a um anexo do Transit Gateway

O exemplo a seguir associa a tabela de rotas do gateway de trânsito especificada ao VPC anexo especificado.

```
aws ec2 associate-transit-gateway-route-table \  
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Saída:

```
{  
  "Association": {  
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",  
    "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
    "ResourceId": "vpc-0065acced4EXAMPLE",  
    "ResourceType": "vpc",  
    "State": "associating"  
  }  
}
```

Para obter mais informações, consulte [Associate a Transit Gateway Route Table](#) no AWS Transit Gateways Guide.

- Para API obter detalhes, consulte [AssociateTransitGatewayRouteTable](#) na Referência de AWS CLI Comandos.

associate-vpc-cidr-block

O código de exemplo a seguir mostra como usar `associate-vpc-cidr-block`.

AWS CLI

Exemplo 1: Para associar um IPv6 CIDR bloco fornecido pela Amazon a um VPC

O `associate-vpc-cidr-block` exemplo a seguir associa um IPv6 CIDR bloco ao especificado VPC. :

```
aws ec2 associate-vpc-cidr-block \  
  --amazon-provided-ipv6-cidr-block \  
  --ipv6-cidr-block-network-border-group us-west-2-lax-1 \  
  --vpc-id vpc-8EXAMPLE
```

Saída:

```
{
```

```

    "Ipv6CidrBlockAssociation": {
      "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",
      "Ipv6CidrBlockState": {
        "State": "associating"
      },
      "NetworkBorderGroup": "us-west-2-lax-1"
    },
    "VpcId": "vpc-8EXAMPLE"
  }

```

Exemplo 2: Para associar um IPv4 CIDR bloco adicional a um VPC

O `associate-vpc-cidr-block` exemplo a seguir associa o IPv4 CIDR bloco `10.2.0.0/16` ao especificado VPC.

```

aws ec2 associate-vpc-cidr-block \
  --vpc-id vpc-1EXAMPLE \
  --cidr-block 10.2.0.0/16

```

Saída:

```

{
  "CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-2EXAMPLE",
    "CidrBlock": "10.2.0.0/16",
    "CidrBlockState": {
      "State": "associating"
    }
  },
  "VpcId": "vpc-1EXAMPLE"
}

```

- Para API obter detalhes, consulte [AssociateVpcCidrBlock](#) na Referência de AWS CLI Comandos.

attach-classic-link-vpc

O código de exemplo a seguir mostra como usar `attach-classic-link-vpc`.

AWS CLI

Para vincular (anexar) uma instância EC2 -Classic a um VPC

Este exemplo vincula a instância `i-1234567890abcdef0` à `vpc-88888888` por meio do grupo de segurança `sg-12312312`. VPC VPC

Comando:

```
aws ec2 attach-classic-link-vpc --instance-id i-1234567890abcdef0 --vpc-id vpc-88888888 --groups sg-12312312
```

Saída:

```
{  
  "Return": true  
}
```

- Para API obter detalhes, consulte [AttachClassicLinkVpc](#) na Referência de AWS CLI Comandos.

attach-internet-gateway

O código de exemplo a seguir mostra como usar `attach-internet-gateway`.

AWS CLI

Para conectar um gateway de internet ao seu VPC

O `attach-internet-gateway` exemplo a seguir anexa o gateway de internet especificado ao específico VPC.

```
aws ec2 attach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Gateways de Internet](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [AttachInternetGateway](#) na Referência de AWS CLI Comandos.

attach-network-interface

O código de exemplo a seguir mostra como usar `attach-network-interface`.

AWS CLI

Exemplo 1: Para conectar uma interface de rede a uma instância

O `attach-network-interface` exemplo a seguir anexa a interface de rede especificada à instância especificada.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-0dc56a8d4640ad10a \  
  --instance-id i-1234567890abcdef0 \  
  --device-index 1
```

Saída:

```
{  
  "AttachmentId": "eni-attach-01a8fc87363f07cf9"  
}
```

Para obter mais informações, consulte [Interfaces de rede elásticas](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para conectar uma interface de rede a uma instância com várias placas de rede

O `attach-network-interface` exemplo a seguir anexa a interface de rede especificada à instância e à placa de rede especificadas.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-07483b1897541ad83 \  
  --instance-id i-01234567890abcdef \  
  --network-card-index 1 \  
  --device-index 1
```

Saída:

```
{  
  "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"  
}
```

Para obter mais informações, consulte [Interfaces de rede elásticas](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [AttachNetworkInterface](#) na Referência de AWS CLI Comandos.

attach-verified-access-trust-provider

O código de exemplo a seguir mostra como usar `attach-verified-access-trust-provider`.

AWS CLI

Para vincular um provedor confiável a uma instância

O `attach-verified-access-trust-provider` exemplo a seguir anexa o provedor confiável de acesso verificado especificado à instância de acesso verificado especificada.

```
aws ec2 attach-verified-access-trust-provider \  
  --verified-access-instance-id vai-0ce000c0b7643abea \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}  
}
```

Para obter mais informações, consulte [Instâncias de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [AttachVerifiedAccessTrustProvider](#) na Referência de AWS CLI Comandos.

attach-volume

O código de exemplo a seguir mostra como usar `attach-volume`.

AWS CLI

Para anexar um volume a uma instância

Este exemplo de comando anexa um volume (`vol-1234567890abcdef0`) a uma instância (`i-01474ef662b89480`) como `/dev/sdf`.

Comando:

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-  
id i-01474ef662b89480 --device /dev/sdf
```

Saída:

```
{  
  "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",  
  "InstanceId": "i-01474ef662b89480",  
  "VolumeId": "vol-1234567890abcdef0",  
  "State": "attaching",  
  "Device": "/dev/sdf"  
}
```

- Para API obter detalhes, consulte [AttachVolume](#) na Referência de AWS CLI Comandos.

attach-vpn-gateway

O código de exemplo a seguir mostra como usar `attach-vpn-gateway`.

AWS CLI

Para conectar um gateway privado virtual ao seu VPC

O `attach-vpn-gateway` exemplo a seguir anexa o gateway privado virtual especificado ao especificadoVPC.

```
aws ec2 attach-vpn-gateway \  
  --vpn-gateway-id vgw-9a4cacf3 \  
  --vpc-id vpc-a01106c2
```

Saída:

```
{  
  "VpcAttachment": {  
    "State": "attaching",  
    "VpcId": "vpc-a01106c2"  
  }  
}
```

- Para API obter detalhes, consulte [AttachVpnGateway](#) na Referência de AWS CLI Comandos.

`authorize-client-vpn-ingress`

O código de exemplo a seguir mostra como usar `authorize-client-vpn-ingress`.

AWS CLI

Para adicionar uma regra de autorização para um VPN endpoint do cliente

O `authorize-client-vpn-ingress` exemplo a seguir adiciona uma regra de autorização de entrada que permite que todos os clientes acessem 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
```

Saída:

```
{  
  "Status": {
```

```
    "Code": "authorizing"  
  }  
}
```

Para obter mais informações, consulte [Regras de autorização](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [AuthorizeClientVpnIngress](#) na Referência de AWS CLI Comandos.

authorize-security-group-egress

O código de exemplo a seguir mostra como usar `authorize-security-group-egress`.

AWS CLI

Para adicionar uma regra que permita tráfego de saída para um intervalo de endereços específico

Este exemplo de comando adiciona uma regra que concede acesso aos intervalos de endereços especificados na 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}}]
```

Para adicionar uma regra que permita tráfego de saída para um grupo de segurança específico

Este exemplo de comando adiciona uma regra que concede acesso ao grupo de segurança especificado na 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}]
```

- Para API obter detalhes, consulte [AuthorizeSecurityGroupEgress](#) na Referência de AWS CLI Comandos.

authorize-security-group-ingress

O código de exemplo a seguir mostra como usar `authorize-security-group-ingress`.

AWS CLI

Exemplo 1: Para adicionar uma regra que permita tráfego de entrada SSH

O `authorize-security-group-ingress` exemplo a seguir adiciona uma regra que permite tráfego de entrada na TCP porta 22 (SSH).

```
aws ec2 authorize-security-group-ingress \  
  --group-id sg-1234567890abcdef0 \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Exemplo 2: Para adicionar uma regra que permita HTTP tráfego de entrada de outro grupo de segurança

O `authorize-security-group-ingress` exemplo a seguir adiciona uma regra que permite acesso de entrada na TCP porta 80 do grupo `sg-1a2b3c4d` de segurança de origem. O grupo de origem deve estar no mesmo grupo VPC ou em um peer VPC (requer uma conexão de VPC peering). O tráfego de entrada é permitido com base nos endereços IP privados das instâncias associadas ao grupo de segurança de origem (e não ao endereço IP público ou ao endereço IP elástico).

```
aws ec2 authorize-security-group-ingress \  
  --group-id sg-1234567890abcdef0 \  
  --protocol tcp \  
  --port 80 \  
  --source-group sg-1a2b3c4d
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Exemplo 3: adicionar várias regras na mesma chamada

O `authorize-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para adicionar duas regras de entrada, uma que habilita o acesso de entrada na TCP porta 3389 (RDP) e a outra que habilita ping/. ICMP

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol =tcp, =3389, FromPort =3389, = "[{=172.31.0.0/16}]" =icmp, =-1, =-1, =
"[{=172.31.0.0/16}ToPort]" IpRanges CidrIp IpProtocol FromPort ToPort IpRanges CidrIp
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 4: Para adicionar uma regra para ICMP tráfego

O `authorize-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para adicionar uma regra de entrada que permite a ICMP mensagem `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (Tipo 3, Código 4) de qualquer lugar.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=icmp, =3, =4, = [{"IpProtocol": "icmp", "FromPort": 3, "ToPort": 4, "IpRanges": [{"CidrIp": "0.0.0.0/0"}]}
```

Saída:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupId": "sgr-0de3811019069b787",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmp",
      "FromPort": 3,
      "ToPort": 4,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

Exemplo 5: Para adicionar uma regra para IPv6 tráfego

O `authorize-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para adicionar uma regra de entrada que permite SSH acesso (porta 22) a partir do IPv6 intervalo `2001:db8:1234:1a00::/64`.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=tcp, =22, =22, Ipv6Ranges= [{"IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "Ipv6Ranges": [{"CidrIpv6": "2001:db8:1234:1a00::/64"}]}
```

Saída:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupId": "sgr-0455bc68b60805563",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "Ipv6Ranges": [{"CidrIpv6": "2001:db8:1234:1a00::/64"}]
    }
  ]
}
```



```

        "FromPort": 22,
        "ToPort": 22,
        "CidrIpv6": "2001:db8:1234:1a00::/64"
    }
]
}

```

Exemplo 6: Para adicionar uma regra para ICMPv6 tráfego

O `authorize-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para adicionar uma regra de entrada que permite ICMPv6 tráfego de qualquer lugar.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions =icmpv6, Ipv6Ranges= "[{6=: :/0}]" IpProtocol CidrIpv
```

Saída:

```

{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmpv6",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv6": "::/0"
    }
  ]
}

```

Exemplo 7: adicionar uma regra com uma descrição

O `authorize-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para adicionar uma regra de entrada que permite o RDP tráfego do intervalo de IPv4 endereços especificado. A regra inclui uma descrição para ajudar a identificá-lo posteriormente.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol =tcp, =3389, =3389, = "[{=203.0.113.0/24, Descrição=' acesso do escritório de Nova York }]" FromPort ToPort IpRanges CidrIpv RDP
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 8: adicionar uma regra de entrada que use uma lista de prefixos

O `authorize-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para adicionar uma regra de entrada que permite todo o tráfego para os CIDR intervalos na lista de prefixos especificada.

```
aws ec2 authorize-security-group-ingress --group-id sg-04a351bfe432d4e71 --ip-permissions =all,
= "[{=pl-002dc3ec097de1514}]" IpProtocol PrefixListIds PrefixListId
```

Saída:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "PrefixListId": "pl-0721453c7ac4ec009"
    }
  ]
}
```

```
]
}
```

Para obter mais informações, consulte [Grupos de segurança](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [AuthorizeSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

bundle-instance

O código de exemplo a seguir mostra como usar `bundle-instance`.

AWS CLI

Para agrupar uma instância

Este exemplo agrupa `i-1234567890abcdef0` a instância em um bucket chamado `bundletasks`. Antes de especificar valores para sua chave de acesso IDs, revise e siga as orientações em [Melhores práticas para gerenciar chaves de AWS acesso](#).

Comando:

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

Saída:

```
{
  "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"
  }
}
```

```
}
```

- Para API obter detalhes, consulte [BundleInstance](#) na Referência de AWS CLI Comandos.

cancel-bundle-task

O código de exemplo a seguir mostra como usar `cancel-bundle-task`.

AWS CLI

Para cancelar uma tarefa do pacote

Este exemplo cancela a tarefa do pacote. `bun-2a4e041c`

Comando:

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CancelBundleTask](#) na Referência de AWS CLI Comandos.

cancel-capacity-reservation-fleets

O código de exemplo a seguir mostra como usar `cancel-capacity-reservation-fleets`.

AWS CLI

Para cancelar uma frota de reserva de capacidade

O `cancel-capacity-reservation-fleets` exemplo a seguir cancela a frota de reserva de capacidade especificada e a capacidade que ela reserva. Quando você cancela uma frota, seu status muda para `cancelled` e ela não pode mais criar novas reservas de capacidade. Além disso, todas as reservas de capacidade individuais na frota são canceladas e as instâncias que estavam sendo executadas anteriormente na capacidade reservada continuam funcionando normalmente em capacidade compartilhada.

```
aws ec2 cancel-capacity-reservation-fleets \  
--capacity-reservation-fleet-ids crf-abcdef01234567890
```

Saída:

```
{  
  "SuccessfulFleetCancellations": [  
    {  
      "CurrentFleetState": "cancelling",  
      "PreviousFleetState": "active",  
      "CapacityReservationFleetId": "crf-abcdef01234567890"  
    }  
  ],  
  "FailedFleetCancellations": []  
}
```

Para obter mais informações sobre frotas de reserva de capacidade, consulte [Frotas de reserva de capacidade](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [CancelCapacityReservationFleets](#) na Referência de AWS CLI Comandos.

cancel-capacity-reservation

O código de exemplo a seguir mostra como usar `cancel-capacity-reservation`.

AWS CLI

Para cancelar uma reserva de capacidade

O `cancel-capacity-reservation` exemplo a seguir cancela a reserva de capacidade especificada.

```
aws ec2 cancel-capacity-reservation \  
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Cancelamento de uma reserva de capacidade](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [CancelCapacityReservation](#) na Referência de AWS CLI Comandos.

cancel-conversion-task

O código de exemplo a seguir mostra como usar `cancel-conversion-task`.

AWS CLI

Para cancelar uma conversão ativa de uma instância ou volume

Este exemplo cancela o upload associado à ID da tarefa `import-i-fh95npoc`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- Para API obter detalhes, consulte [CancelConversionTask](#) na Referência de AWS CLI Comandos.

cancel-export-task

O código de exemplo a seguir mostra como usar `cancel-export-task`.

AWS CLI

Para cancelar uma tarefa de exportação ativa

Este exemplo cancela uma tarefa de exportação ativa com o ID de tarefa `export-i-fgelt0i7`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- Para API obter detalhes, consulte [CancelExportTask](#) na Referência de AWS CLI Comandos.

cancel-image-launch-permission

O código de exemplo a seguir mostra como usar `cancel-image-launch-permission`.

AWS CLI

Para cancelar o AMI compartilhamento com sua conta da Amazon Web Services

O `cancel-image-launch-permission` exemplo a seguir remove sua conta das permissões AMI de lançamento especificadas.

```
aws ec2 cancel-image-launch-permission \  
  --image-id ami-0123456789example \  
  --region us-east-1
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Cancelar o AMI compartilhamento com sua conta da Amazon Web Services](#) no Guia EC2 do Usuário da Amazon.

- Para API obter detalhes, consulte [CancelImageLaunchPermission](#) na Referência de AWS CLI Comandos.

cancel-import-task

O código de exemplo a seguir mostra como usar `cancel-import-task`.

AWS CLI

Para cancelar uma tarefa de importação

O `cancel-import-task` exemplo a seguir cancela a tarefa de importação de imagem especificada.

```
aws ec2 cancel-import-task \  
  --import-task-id import-ami-1234567890abcdef0
```

Saída:

```
{  
  "ImportTaskId": "import-ami-1234567890abcdef0",  
  "PreviousState": "active",  
  "State": "deleting"  
}
```

- Para API obter detalhes, consulte [CancelImportTask](#) na Referência de AWS CLI Comandos.

cancel-reserved-instances-listing

O código de exemplo a seguir mostra como usar `cancel-reserved-instances-listing`.

AWS CLI

Para cancelar uma listagem de Instâncias Reservadas

O `cancel-reserved-instances-listing` exemplo a seguir cancela a listagem de Instâncias Reservadas especificada.

```
aws ec2 cancel-reserved-instances-listing \  
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- Para API obter detalhes, consulte [CancelReservedInstancesListing](#) na Referência de AWS CLI Comandos.

cancel-spot-fleet-requests

O código de exemplo a seguir mostra como usar `cancel-spot-fleet-requests`.

AWS CLI

Exemplo 1: Para cancelar uma solicitação de frota spot e encerrar as instâncias associadas

O `cancel-spot-fleet-requests` exemplo a seguir cancela uma solicitação de frota spot e encerra as instâncias sob demanda e as instâncias spot associadas.

```
aws ec2 cancel-spot-fleet-requests \
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --terminate-instances
```

Saída:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_terminating",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

Para obter mais informações, consulte [Cancelar uma solicitação de frota spot](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: Para cancelar uma solicitação de frota spot sem encerrar as instâncias associadas

O `cancel-spot-fleet-requests` exemplo a seguir cancela uma solicitação de frota spot sem encerrar as instâncias sob demanda e as instâncias spot associadas.

```
aws ec2 cancel-spot-fleet-requests \
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
  --no-terminate-instances
```

Saída:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_running",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

Para obter mais informações, consulte [Cancelar uma solicitação de frota spot](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [CancelSpotFleetRequests](#) na Referência de AWS CLI Comandos.

cancel-spot-instance-requests

O código de exemplo a seguir mostra como usar `cancel-spot-instance-requests`.

AWS CLI

Para cancelar solicitações de Instância Spot

Este exemplo de comando cancela uma solicitação de instância spot.

Comando:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Saída:

```
{
  "CancelledSpotInstanceRequests": [
    {
      "State": "cancelled",
      "SpotInstanceRequestId": "sir-08b93456"
    }
  ]
}
```

- Para API obter detalhes, consulte [CancelSpotInstanceRequests](#) na Referência de AWS CLI Comandos.

confirm-product-instance

O código de exemplo a seguir mostra como usar `confirm-product-instance`.

AWS CLI

Para confirmar a instância do produto

Esse exemplo determina se o código do produto especificado está associado à instância especificada.

Comando:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-id i-1234567890abcdef0
```

Saída:

```
{  
  "OwnerId": "123456789012"  
}
```

- Para API obter detalhes, consulte [ConfirmProductInstance](#) na Referência de AWS CLI Comandos.

copy-fpga-image

O código de exemplo a seguir mostra como usar `copy-fpga-image`.

AWS CLI

Para copiar uma FPGA imagem da Amazon

Este exemplo copia o especificado AFI da `us-east-1` região para a região atual (`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
```

Saída:

```
{
  "FpgaImageId": "afi-06b12350a123fbabc"
}
```

- Para API obter detalhes, consulte [CopyFpgaImage](#) na Referência de AWS CLI Comandos.

copy-image

O código de exemplo a seguir mostra como usar `copy-image`.

AWS CLI

Exemplo 1: Para copiar um AMI para outra região

O comando de `copy-image` exemplo a seguir copia o especificado AMI da `us-west-2` Região para a `us-east-1` Região e adiciona uma breve descrição.

```
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."
```

Saída:

```
{
  "ImageId": "ami-0123456789abcdefg"
}
```

Para obter mais informações, consulte [Copiar um AMI](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para copiar um AMI para outra região e criptografar o instantâneo de backup

O `copy-image` comando a seguir copia o especificado AMI da `us-west-2` região para a região atual e criptografa o instantâneo de backup usando a chave especificada. 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
```

Saída:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Para obter mais informações, consulte [Copiar um AMI](#) no Guia EC2 do usuário da Amazon.

Exemplo 3: Para incluir suas AMI tags definidas pelo usuário ao copiar um AMI

O `copy-image` comando a seguir usa o `--copy-image-tags` parâmetro para copiar suas AMI tags definidas pelo usuário ao copiar o 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
```

Saída:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Para obter mais informações, consulte [Copiar um AMI](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [CopyImage](#) na Referência de AWS CLI Comandos.

copy-snapshot

O código de exemplo a seguir mostra como usar `copy-snapshot`.

AWS CLI

Exemplo 1: Para copiar um instantâneo para outra região

O comando de `copy-snapshot` exemplo a seguir copia o instantâneo especificado da `us-west-2` região para a `us-east-1` região e adiciona uma breve descrição.

```
aws ec2 copy-snapshot \  
  --region us-east-1 \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --description "This is my copied snapshot."
```

Saída:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Para obter mais informações, consulte [Copiar um EBS snapshot da Amazon](#) no Guia do EC2 usuário da Amazon.

Exemplo 2: Para copiar um instantâneo não criptografado e criptografar o novo instantâneo

O `copy-snapshot` comando a seguir copia o instantâneo não criptografado especificado da `us-west-2` região para a região atual e criptografa o novo instantâneo usando a chave especificada. KMS

```
aws ec2 copy-snapshot \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

Saída:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Para obter mais informações, consulte [Copiar um EBS snapshot da Amazon](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [CopySnapshot](#) na Referência de AWS CLI Comandos.

create-capacity-reservation-fleet

O código de exemplo a seguir mostra como usar `create-capacity-reservation-fleet`.

AWS CLI

Para criar uma frota de reserva de capacidade

O `create-capacity-reservation-fleet` exemplo a seguir cria uma frota de reserva de capacidade para o tipo de instância especificado na solicitação, até a capacidade total especificada. O número de instâncias para as quais a frota de reserva de capacidade reserva capacidade depende da capacidade total de destino e dos pesos de tipo de instância que você especifica na solicitação. Especifique os tipos de instância a serem usados e uma prioridade para cada um dos tipos de instância designados.

```
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
```

Conteúdo de `instanceTypeSpecification.json`:

```
[  
  {  
    "InstanceType": "m5.xlarge",  
    "InstancePlatform": "Linux/UNIX",  
    "Weight": 3.0,  
    "AvailabilityZone": "us-east-1a",  
    "EbsOptimized": true,  
    "Priority" : 1  
  }  
]
```

Saída:

```
{
  "Status": "submitted",
  "TotalFulfilledCapacity": 0.0,
  "CapacityReservationFleetId": "crf-abcdef01234567890",
  "TotalTargetCapacity": 24
}
```

Para obter mais informações sobre frotas de reserva de capacidade, consulte [Frotas de reserva de capacidade](#) no Guia EC2 do usuário da Amazon.

Para obter mais informações sobre o peso do tipo de instância e a capacidade total alvo, consulte [Peso do tipo de instância](#) e [Capacidade total alvo](#) no Guia EC2 do usuário da Amazon.

Para obter mais informações sobre a designação de prioridade para tipos de instância específicos, consulte [Estratégia de alocação](#) e [Prioridade do tipo de instância](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [CreateCapacityReservationFleet](#) na Referência de AWS CLI Comandos.

create-capacity-reservation

O código de exemplo a seguir mostra como usar create-capacity-reservation.

AWS CLI

Exemplo 1: Para criar uma reserva de capacidade

O create-capacity-reservation exemplo a seguir cria uma reserva de capacidade na zona de eu-west-1a disponibilidade, na qual você pode executar três t2.medium instâncias executando um sistema operacional Linux/Unix. Por padrão, a reserva de capacidade é criada com critérios de correspondência de instâncias abertas e sem suporte para armazenamento temporário, e permanece ativa até que você a cancele manualmente.

```
aws ec2 create-capacity-reservation \
  --availability-zone eu-west-1a \
  --instance-type t2.medium \
  --instance-platform Linux/UNIX \
  --instance-count 3
```

Saída:


```
{
  "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"
  }
}
```

Exemplo 2: Para criar uma reserva de capacidade que termine automaticamente em uma data/hora especificada

O `create-capacity-reservation` exemplo a seguir cria uma reserva de capacidade na zona de `eu-west-1a` disponibilidade, na qual você pode executar três `m5.large` instâncias executando um sistema operacional Linux/Unix. Essa reserva de capacidade termina automaticamente em 31/08/2019 às 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
```

Saída:

```
{
  "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"
  }
}

```

Exemplo 3: Para criar uma reserva de capacidade que aceite somente inicializações de instâncias específicas

O `create-capacity-reservation` exemplo a seguir cria uma reserva de capacidade que aceita somente lançamentos de instâncias direcionadas.

```

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

```

Saída:

```

{
  "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"
  }
}

```

```
}  
}
```

Para obter mais informações, consulte [Criação de uma reserva de capacidade](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [CreateCapacityReservation](#) na Referência de AWS CLI Comandos.

create-carrier-gateway

O código de exemplo a seguir mostra como usar `create-carrier-gateway`.

AWS CLI

Para criar um gateway de operadora

O `create-carrier-gateway` exemplo a seguir cria um gateway de operadora para o especificado VPC.

```
aws ec2 create-carrier-gateway \  
  --vpc-id vpc-0c529aEXAMPLE1111
```

Saída:

```
{  
  "CarrierGateway": {  
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",  
    "VpcId": "vpc-0c529aEXAMPLE1111",  
    "State": "pending",  
    "OwnerId": "123456789012"  
  }  
}
```

Para obter mais informações, consulte [Carrier gateways](#) no AWS Wavelength User Guide.

- Para API obter detalhes, consulte [CreateCarrierGateway](#) na Referência de AWS CLI Comandos.

create-client-vpn-endpoint

O código de exemplo a seguir mostra como usar `create-client-vpn-endpoint`.

AWS CLI

Para criar um VPN endpoint de cliente

O `create-client-vpn-endpoint` exemplo a seguir cria um VPN endpoint do cliente que usa autenticação mútua e especifica um valor para o bloco do clienteCIDR.

```
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
```

Saída:

```
{  
  "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",  
  "Status": {  
    "Code": "pending-associate"  
  },  
  "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-  
south-1.amazonaws.com"  
}
```

Para obter mais informações, consulte [Client VPN Endpoints](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [CreateClientVpnEndpoint](#) na Referência de AWS CLI Comandos.

`create-client-vpn-route`

O código de exemplo a seguir mostra como usar `create-client-vpn-route`.

AWS CLI

Para criar uma rota para um VPN endpoint do cliente

O `create-client-vpn-route` exemplo a seguir adiciona uma rota à internet (`0.0.0.0/0`) para a sub-rede especificada do VPN endpoint do cliente.

```
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
```

Saída:

```
{  
  "Status": {  
    "Code": "creating"  
  }  
}
```

Para obter mais informações, consulte [Rotas](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [CreateClientVpnRoute](#) na Referência de AWS CLI Comandos.

create-coip-cidr

O código de exemplo a seguir mostra como usar `create-coip-cidr`.

AWS CLI

Para criar uma variedade de endereços IP de propriedade do cliente (CoIP)

O `create-coip-cidr` exemplo a seguir cria o intervalo especificado de endereços de CoIP no pool de CoIP especificado.

```
aws ec2 create-coip-cidr \  
  --cidr 15.0.0.0/24 \  
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Saída:

```
{  
  "CoipCidr": {  
    "Cidr": "15.0.0.0/24",  
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
```

```
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"  
  }  
}
```

Para obter mais informações, consulte [Endereços IP pertencentes ao cliente](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [CreateCoipCidr](#) na Referência de AWS CLI Comandos.

create-coip-pool

O código de exemplo a seguir mostra como usar `create-coip-pool`.

AWS CLI

Para criar um pool de endereços IP de propriedade do cliente (CoIP)

O `create-coip-pool` exemplo a seguir cria um pool de CoIP para endereços de CoIP na tabela de rotas do gateway local especificada.

```
aws ec2 create-coip-pool \  
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Saída:

```
{  
  "CoipPool": {  
    "PoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-  
coip-1234567890abcdefg"  
  }  
}
```

Para obter mais informações, consulte [Endereços IP pertencentes ao cliente](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [CreateCoipPool](#) na Referência de AWS CLI Comandos.

create-customer-gateway

O código de exemplo a seguir mostra como usar `create-customer-gateway`.

AWS CLI

Para criar um gateway do cliente

Este exemplo cria um gateway de cliente com o endereço IP especificado para sua interface externa.

Comando:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

Saída:

```
{
  "CustomerGateway": {
    "CustomerGatewayId": "cgw-0e11f167",
    "IpAddress": "12.1.2.3",
    "State": "available",
    "Type": "ipsec.1",
    "BgpAsn": "65534"
  }
}
```

- Para API obter detalhes, consulte [CreateCustomerGateway](#) na Referência de AWS CLI Comandos.

create-default-subnet

O código de exemplo a seguir mostra como usar `create-default-subnet`.

AWS CLI

Para criar uma sub-rede padrão

Este exemplo cria uma sub-rede padrão na Zona `us-east-2a` de Disponibilidade.

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
}
}
```

- Para API obter detalhes, consulte [CreateDefaultSubnet](#) na Referência de AWS CLI Comandos.

create-default-vpc

O código de exemplo a seguir mostra como usar `create-default-vpc`.

AWS CLI

Para criar um padrão VPC

Este exemplo cria um padrão VPC.

Comando:

```
aws ec2 create-default-vpc
```

Saída:

```
{
  "Vpc": {
    "VpcId": "vpc-8eaae5ea",
    "InstanceTenancy": "default",
    "Tags": [],
    "Ipv6CidrBlockAssociationSet": [],
    "State": "pending",
    "DhcpOptionsId": "dopt-af0c32c6",
    "CidrBlock": "172.31.0.0/16",
```



```
    "IsDefault": true
  }
}
```

- Para API obter detalhes, consulte [CreateDefaultVpc](#) na Referência de AWS CLI Comandos.

create-dhcp-options

O código de exemplo a seguir mostra como usar `create-dhcp-options`.

AWS CLI

Para criar um conjunto de DHCP opções

O `create-dhcp-options` exemplo a seguir cria um conjunto de DHCP opções que especifica o nome do domínio, os servidores de nomes de domínio e o tipo de BIOS nó da rede.

```
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"
```

Saída:

```
{  
  "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"
}
}

```

- Para API obter detalhes, consulte [CreateDhcpOptions](#) na Referência de AWS CLI Comandos.

create-egress-only-internet-gateway

O código de exemplo a seguir mostra como usar `create-egress-only-internet-gateway`.

AWS CLI

Para criar um gateway de Internet somente de saída

Este exemplo cria um gateway de Internet somente de saída para o especificado. VPC

Comando:

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

Saída:

```

{
  "EgressOnlyInternetGateway": {
    "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
    "Attachments": [
      {
        "State": "attached",

```

```

        "VpcId": "vpc-0c62a468"
    }
  ]
}
}

```

- Para API obter detalhes, consulte [CreateEgressOnlyInternetGateway](#) na Referência de AWS CLI Comandos.

create-fleet

O código de exemplo a seguir mostra como usar `create-fleet`.

AWS CLI

Exemplo 1: Para criar uma EC2 frota que lança instâncias spot como modelo de compra padrão

O `create-fleet` exemplo a seguir cria uma EC2 frota usando os parâmetros mínimos necessários para lançar uma frota: um modelo de lançamento, capacidade alvo e modelo de compra padrão. O modelo de execução é identificado pelo ID do seu modelo de execução e o número da versão. A capacidade alvo da frota é de 2 instâncias, e o modelo de compra padrão é spot, o que resulta no lançamento de 2 instâncias spot da frota.

Ao criar uma EC2 frota, use um JSON arquivo para especificar informações sobre as instâncias a serem executadas.

```

aws ec2 create-fleet \
  --cli-input-json file://file_name.json

```

Conteúdo do `file_name.json`:

```

{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {

```

```
    "TotalTargetCapacity": 2,  
    "DefaultTargetCapacityType": "spot"  
  }  
}
```

Saída:

```
{  
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

Exemplo 2: Para criar uma EC2 frota que lança instâncias sob demanda como modelo de compra padrão

O `create-fleet` exemplo a seguir cria uma EC2 frota usando os parâmetros mínimos necessários para lançar uma frota: um modelo de lançamento, capacidade alvo e modelo de compra padrão. O modelo de execução é identificado pelo ID do seu modelo de execução e o número da versão. A capacidade alvo da frota é de 2 instâncias, e o modelo de compra padrão é `on-demand`, o que resulta no lançamento de 2 instâncias sob demanda da frota.

Ao criar uma EC2 frota, use um JSON arquivo para especificar informações sobre as instâncias a serem executadas.

```
aws ec2 create-fleet \  
  --cli-input-json file://file_name.json
```

Conteúdo do `file_name.json`:

```
{  
  "LaunchTemplateConfigs": [  
    {  
      "LaunchTemplateSpecification": {  
        "LaunchTemplateId": "lt-0e8c754449b27161c",  
        "Version": "1"  
      }  
    }  
  ],  
  "TargetCapacitySpecification": {  
    "TotalTargetCapacity": 2,  
    "DefaultTargetCapacityType": "on-demand"  
  }  
}
```

```
}  
}
```

Saída:

```
{  
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

Exemplo 3: Para criar uma EC2 frota que lance instâncias sob demanda como capacidade primária

O `create-fleet` exemplo a seguir cria uma EC2 frota que especifica a capacidade alvo total de 2 instâncias para a frota e uma capacidade alvo de 1 instância sob demanda. O modelo de compra padrão é spot. A frota inicia 1 instância sob demanda conforme especificado, mas precisa iniciar mais uma instância para atingir a capacidade total pretendida. O modelo de compra da diferença é calculado como `TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType`, o que faz com que a frota lance 1 instância spot.

Ao criar uma EC2 frota, use um JSON arquivo para especificar informações sobre as instâncias a serem executadas.

```
aws ec2 create-fleet \  
  --cli-input-json file://file_name.json
```

Conteúdo do `file_name.json`:

```
{  
  "LaunchTemplateConfigs": [  
    {  
      "LaunchTemplateSpecification": {  
        "LaunchTemplateId": "lt-0e8c754449b27161c",  
        "Version": "1"  
      }  
    }  
  ],  
  "TargetCapacitySpecification": {  
    "TotalTargetCapacity": 2,  
    "OnDemandTargetCapacity": 1,  
    "DefaultTargetCapacityType": "spot"  
  }  
}
```

```
}  
}
```

Saída:

```
{  
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

Exemplo 4: Para criar uma EC2 frota que lança instâncias spot usando a estratégia de alocação de menor preço

Se a estratégia de alocação para Instâncias spot não for especificada, a estratégia de alocação padrão, `lowest-price`, será usada. O `create-fleet` exemplo a seguir cria uma EC2 frota usando a estratégia de `lowest-price` alocação. As três especificações de execução, que substituem o modelo de execução, têm tipos de instância diferentes, mas a mesma capacidade ponderada e sub-rede. A capacidade total pretendida é de 2 instâncias e o modelo de compra padrão é spot. A EC2 frota lança 2 instâncias spot usando o tipo de instância da especificação de lançamento com o menor preço.

Ao criar uma EC2 frota, use um JSON arquivo para especificar informações sobre as instâncias a serem executadas.

```
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"
}
}

```

Saída:

```

{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}

```

- Para API obter detalhes, consulte [CreateFleet](#) na Referência de AWS CLI Comandos.

create-flow-logs

O código de exemplo a seguir mostra como usar `create-flow-logs`.

AWS CLI

Exemplo 1: Para criar um registro de fluxo

O `create-flow-logs` exemplo a seguir cria um registro de fluxo que captura todo o tráfego rejeitado para a interface de rede especificada. Os registros de fluxo são entregues a um grupo de CloudWatch registros no Logs usando as permissões na IAM função especificada.

```

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
```

Saída:

```
{  
  "ClientToken": "so0eNA2uSHUN1HI0S2cJ305GuIX1CezaRdGtexample",  
  "FlowLogIds": [  
    "fl-12345678901234567"  
  ],  
  "Unsuccessful": []  
}
```

Para obter mais informações, consulte [Logs de VPC fluxo](#) no Guia VPC do usuário da Amazon.

Exemplo 2: Para criar um registro de fluxo com um formato personalizado

O `create-flow-logs` exemplo a seguir cria um log de fluxo que captura todo o tráfego para o especificado VPC e entrega os logs de fluxo para um bucket do Amazon S3. O parâmetro `--log-format` especifica um formato personalizado para os registros de log de fluxo. Para executar esse comando no Windows, altere as aspas simples (') para aspas duplas (").

```
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}'
```

Para obter mais informações, consulte [Logs de VPC fluxo](#) no Guia VPC do usuário da Amazon.

Exemplo 3: Para criar um registro de fluxo com um intervalo máximo de agregação de um minuto

O `create-flow-logs` exemplo a seguir cria um log de fluxo que captura todo o tráfego para o especificado VPC e entrega os logs de fluxo para um bucket do Amazon S3. O `--max-aggregation-interval` parâmetro especifica um intervalo máximo de agregação de 60 segundos (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
```

Para obter mais informações, consulte [Logs de VPC fluxo](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [CreateFlowLogs](#) na Referência de AWS CLI Comandos.

create-fpga-image

O código de exemplo a seguir mostra como usar `create-fpga-image`.

AWS CLI

Para criar uma FPGA imagem da Amazon

Este exemplo cria um a AFI partir do tarball especificado no bucket especificado.

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
```

Saída:

```
{  
  "FpgaImageId": "afi-0d123e123bfc85abc",  
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"  
}
```

- Para API obter detalhes, consulte [CreateFpgaImage](#) na Referência de AWS CLI Comandos.

create-image

O código de exemplo a seguir mostra como usar `create-image`.

AWS CLI

Exemplo 1: Para criar um a AMI partir de uma instância EBS apoiada pela Amazon

O `create-image` exemplo a seguir cria um AMI a partir da instância especificada.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --description "An AMI for my server"
```

Saída:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Para obter mais informações sobre como especificar um mapeamento de dispositivos de blocos para vocêAMI, consulte [Especificação de um mapeamento de dispositivos de blocos para um AMI no Guia EC2](#) do usuário da Amazon.

Exemplo 2: Para criar um a AMI partir de uma instância EBS apoiada pela Amazon sem reinicializar

O `create-image` exemplo a seguir cria AMI e define o parâmetro `--no-reboot`, para que a instância não seja reinicializada antes da criação da imagem.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --no-reboot
```

Saída:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Para obter mais informações sobre como especificar um mapeamento de dispositivos de blocos para vocêAMI, consulte [Especificação de um mapeamento de dispositivos de blocos para um AMI no Guia EC2](#) do usuário da Amazon.

Exemplo 3: Para marcar um AMI e tirar fotos na criação

O `create-image` exemplo a seguir cria um AMI, e marca o AMI e os instantâneos com a mesma 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}]"
```

Saída:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Para obter mais informações sobre como marcar seus recursos na criação, consulte [Adicionar tags na criação de recursos](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [CreateImage](#) na Referência de AWS CLI Comandos.

create-instance-connect-endpoint

O código de exemplo a seguir mostra como usar `create-instance-connect-endpoint`.

AWS CLI

Para criar um endpoint do EC2 Instance Connect

O `create-instance-connect-endpoint` exemplo a seguir cria um endpoint do EC2 Instance Connect na sub-rede especificada.

```
aws ec2 create-instance-connect-endpoint \  
  --region us-east-1 \  
  --subnet-id subnet-0123456789example
```

Saída:

```
{
```

```

    "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"
}

```

Para obter mais informações, consulte [Create an EC2 Instance Connect Endpoint](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [CreateInstanceConnectEndpoint](#) na Referência de AWS CLI Comandos.

create-instance-event-window

O código de exemplo a seguir mostra como usar `create-instance-event-window`.

AWS CLI

Exemplo 1: Para criar uma janela de eventos com um intervalo de tempo

O `create-instance-event-window` exemplo a seguir cria uma janela de evento com um intervalo de tempo. Você também deve especificar o parâmetro `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

```

Saída:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "TimeRanges": [
      {
        "StartWeekDay": "monday",
        "StartHour": 2,
        "EndWeekDay": "wednesday",
        "EndHour": 8
      }
    ],
    "Name": "myEventWindowName",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}

```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 2: Para criar uma janela de eventos com uma expressão cron

O `create-instance-event-window` exemplo a seguir cria uma janela de eventos com uma expressão cron. Você também deve especificar o parâmetro `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
```

Saída:

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [CreateInstanceEventWindow](#) na Referência de AWS CLI Comandos.

create-instance-export-task

O código de exemplo a seguir mostra como usar `create-instance-export-task`.

AWS CLI

Para exportar uma instância

Esse exemplo de comando cria uma tarefa para exportar a instância `i-1234567890abcdef0` para o bucket `myexportbucket` do Amazon S3.

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
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateInstanceExportTask](#) na Referência de AWS CLI Comandos.

create-internet-gateway

O código de exemplo a seguir mostra como usar `create-internet-gateway`.

AWS CLI

Para criar um gateway de internet

O `create-internet-gateway` exemplo a seguir cria um gateway de internet com a `tagName=my-igw`.

```
aws ec2 create-internet-gateway \
  --tag-specifications ResourceType=internet-gateway,Tags=[{Key=Name,Value=my-igw}]
```

Saída:

```
{
```

```
"InternetGateway": {
  "Attachments": [],
  "InternetGatewayId": "igw-0d0fb496b3994d755",
  "OwnerId": "123456789012",
  "Tags": [
    {
      "Key": "Name",
      "Value": "my-igw"
    }
  ]
}
```

Para obter mais informações, consulte [Gateways de Internet](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [CreateInternetGateway](#) na Referência de AWS CLI Comandos.

create-ipam-pool

O código de exemplo a seguir mostra como usar `create-ipam-pool`.

AWS CLI

Para criar um IPAM pool

O `create-ipam-pool` exemplo a seguir cria um 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"}]
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

```

    ]
  }
}

```

Para obter mais informações, consulte [Plano de provisionamento de endereços IP no Guia VPC](#) IPAMdo usuário da Amazon.

- Para API obter detalhes, consulte [CreatelpamPool](#)na Referência de AWS CLI Comandos.

create-ipam-resource-discovery

O código de exemplo a seguir mostra como usar `create-ipam-resource-discovery`.

AWS CLI

Para criar uma descoberta de recursos

Neste exemplo, você é um IPAM administrador delegado que deseja criar e compartilhar uma descoberta de recursos com o IPAM administrador de outra AWS organização para que o administrador da outra organização possa gerenciar e monitorar os endereços IP dos recursos em sua organização.

Importante

Este exemplo inclui as `--operating-regions` opções `--region` e porque, embora sejam opcionais, elas devem ser configuradas de uma maneira específica para integrar com êxito uma descoberta de recursos a um.* `--operating-regions` deve corresponder às regiões em que você tem recursos que IPAM deseja descobrir. IPAM Se houver regiões nas quais você não IPAM deseja gerenciar os endereços IP (por exemplo, por motivos de conformidade), não os inclua. * `--region` deve corresponder à região de origem da região à qual IPAM você deseja associá-lo. Você deve criar a descoberta de recursos na mesma região em que IPAM foi criada. Por exemplo, se a opção à IPAM qual você está se associando foi criada em `us-east-1`, `--region us-east-1` inclua na solicitação. As opções “`--operating-regionse`” assumem como padrão `--region` a região em que você está executando o comando, caso não as especifique.

Neste exemplo, as regiões operacionais do IPAM que estamos integrando incluem `us-west-1``us-west-2`, e. `ap-south-1` Quando criamos a descoberta de recursos, queremos IPAM descobrir os endereços IP dos recursos em `us-west-1` `us-west-2`, mas não`ap-south-1`. Portanto, estamos incluindo apenas `--operating-regions RegionName='us-west-1' RegionName='us-west-2'` na solicitação.

O `create-ipam-resource-discovery` exemplo a seguir cria uma descoberta IPAM de recursos.

```
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
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Depois de criar uma descoberta de recursos, talvez você queira compartilhá-la com outro administrador IPAM delegado, o que você pode fazer com [create-resource-share](#). Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [CreateIpamResourceDiscovery](#) na Referência de AWS CLI Comandos.

create-ipam-scope

O código de exemplo a seguir mostra como usar create-ipam-scope.

AWS CLI

Para criar um IPAM escopo

O create-ipam-scope exemplo a seguir cria um IPAM escopo.

(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"}]
```

Saída:

```
{  
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Criar escopos adicionais](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [CreateIpamScope](#) na Referência de AWS CLI Comandos.

create-ipam

O código de exemplo a seguir mostra como usar create-ipam.

AWS CLI

Para criar um IPAM

O create-ipam exemplo a seguir cria um 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}]

```

Saída:

```

{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Criar um IPAM](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [CreateIpam](#) na Referência de AWS CLI Comandos.

create-key-pair

O código de exemplo a seguir mostra como usar `create-key-pair`.

AWS CLI

Para criar um par de chaves

Este exemplo cria um par de chaves denominado `MyKeyPair`.

Comando:

```
aws ec2 create-key-pair --key-name MyKeyPair
```

A saída é uma ASCII versão da chave privada e da impressão digital da chave. Você precisa salvar a chave em um arquivo.

Para obter mais informações, consulte Using Key Pairs no Guia do usuário da AWS Command Line Interface.

- Para API obter detalhes, consulte [CreateKeyPair](#) na Referência de AWS CLI Comandos.

create-launch-template-version

O código de exemplo a seguir mostra como usar create-launch-template-version.

AWS CLI

Para criar uma versão do modelo de lançamento

Este exemplo cria uma nova versão do modelo de lançamento com base na versão 1 do modelo de lançamento e especifica uma AMI ID diferente.

Comando:

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
--version-description WebVersion2 --source-version 1 --launch-template-data
'{"ImageId":"ami-c998b6b2"}'
```

Saída:

```
{
  "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"
}
}

```

- Para API obter detalhes, consulte [CreateLaunchTemplateVersion](#) na Referência de AWS CLI Comandos.

create-launch-template

O código de exemplo a seguir mostra como usar `create-launch-template`.

AWS CLI

Exemplo 1: criar um modelo de lançamento

O `create-launch-template` exemplo a seguir cria um modelo de execução que especifica a sub-rede na qual iniciar a instância, atribui um endereço IP público e um IPv6 endereço à instância e cria uma tag para a instância.

```

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"}]}]}'

```

Saída:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-01238c059e3466abc",
    "LaunchTemplateName": "TemplateForWebServer",
    "DefaultVersionNumber": 1,

```



```

    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-01-27T09:13:24.000Z"
  }
}

```

Para obter mais informações, consulte Execução de uma instância em um modelo de execução no Guia do usuário do Amazon Elastic Compute Cloud. Para obter informações sobre como citar parâmetros JSON formatados, consulte Citar cadeias de caracteres no Guia do usuário da interface de linha de AWS comando.

Exemplo 2: Para criar um modelo de lançamento para o Amazon EC2 Auto Scaling

O `create-launch-template` exemplo a seguir cria um modelo de execução com várias tags e um mapeamento de dispositivos de blocos para especificar um EBS volume adicional quando uma instância é executada. Especifique um valor `Groups` que corresponda aos grupos de segurança nos VPC quais seu grupo de Auto Scaling iniciará instâncias. Especifique as sub-redes VPC e como propriedades do grupo 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

```

Saída:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0123c79c33a54e0abc",
    "LaunchTemplateName": "TemplateForAutoScaling",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-04-30T18:16:06.000Z"
  }
}

```

```
}  
}
```

Para obter mais informações, consulte Criação de um modelo de lançamento para um grupo de Auto Scaling no Guia do usuário do Amazon Auto EC2 Scaling. Para obter informações sobre como citar parâmetros JSON formatados, consulte Citar cadeias de caracteres no Guia do usuário da interface de linha de AWS comando.

Exemplo 3: Para criar um modelo de lançamento que especifique a criptografia de volumes EBS

O `create-launch-template` exemplo a seguir cria um modelo de execução que inclui EBS volumes criptografados criados a partir de um snapshot não criptografado. Ele também coloca tags nos volumes durante a criação. Se a criptografia por padrão estiver desabilitada, você deve especificar a opção `"Encrypted"` conforme mostrado no exemplo a seguir. Se você usar a `"KmsKeyId"` opção para especificar um cliente gerenciadoCMK, também deverá especificar a `"Encrypted"` opção, mesmo que a criptografia esteja ativada por padrão.

```
aws ec2 create-launch-template \  
  --launch-template-name TemplateForEncryption \  
  --launch-template-data file://config.json
```

Conteúdo de `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"
            }
        ]
    }
]
}

```

Saída:

```

{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
    "LaunchTemplateName": "TemplateForEncryption",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2020-01-07T19:08:36.000Z"
  }
}

```

Para obter mais informações, consulte [Restauração de um EBS volume da Amazon a partir de um snapshot e criptografia por padrão](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [CreateLaunchTemplate](#) na Referência de AWS CLI Comandos.

create-local-gateway-route-table-virtual-interface-group-association

O código de exemplo a seguir mostra como usar `create-local-gateway-route-table-virtual-interface-group-association`.

AWS CLI

Para associar uma tabela de rotas de gateway local a um grupo de interfaces virtuais (VIFs)

O `create-local-gateway-route-table-virtual-interface-group-association` exemplo a seguir cria uma associação entre a tabela e o VIF grupo de rotas do gateway local especificados.

```
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
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [associações de VIF grupos](#) no Guia do Usuário do AWS Outposts.

- Para API obter detalhes, consulte [CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#) na Referência de AWS CLI Comandos.

create-local-gateway-route-table-vpc-association

O código de exemplo a seguir mostra como usar `create-local-gateway-route-table-vpc-association`.

AWS CLI

Para associar a VPC a a uma tabela de rotas

O `create-local-gateway-route-table-vpc-association` exemplo a seguir associa o especificado VPC à tabela de rotas do gateway local especificada.

```
aws ec2 create-local-gateway-route-table-vpc-association \
```

```
--local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \  
--vpc-id vpc-07ef66ac71EXAMPLE
```

Saída:

```
{  
  "LocalGatewayRouteTableVpcAssociation": {  
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",  
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",  
    "VpcId": "vpc-07ef66ac71EXAMPLE",  
    "State": "associated"  
  }  
}
```

- Para API obter detalhes, consulte [CreateLocalGatewayRouteTableVpcAssociation](#) na Referência de AWS CLI Comandos.

create-local-gateway-route-table

O código de exemplo a seguir mostra como usar `create-local-gateway-route-table`.

AWS CLI

Para criar uma tabela de rotas de gateway local

O `create-local-gateway-route-table` exemplo a seguir cria uma tabela de rotas de gateway local com o modo de VPC roteamento direto.

```
aws ec2 create-local-gateway-route-table \  
--local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \  
--mode direct-vpc-routing
```

Saída:

```
{  
  "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"
  }
}

```

[Para obter mais informações, consulte Tabelas de rotas do gateway local](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [CreateLocalGatewayRouteTable](#) na Referência de AWS CLI Comandos.

create-local-gateway-route

O código de exemplo a seguir mostra como usar `create-local-gateway-route`.

AWS CLI

Para criar uma rota estática para uma tabela de rotas de gateway local

O `create-local-gateway-route` exemplo a seguir cria a rota especificada na tabela de rotas do gateway local especificada.

```

aws ec2 create-local-gateway-route \
  --destination-cidr-block 0.0.0.0/0 \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE

```

Saída:

```

{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
  }
}

```

```
}
```

- Para API obter detalhes, consulte [CreateLocalGatewayRoute](#) na Referência de AWS CLI Comandos.

create-managed-prefix-list

O código de exemplo a seguir mostra como usar `create-managed-prefix-list`.

AWS CLI

Para criar uma lista de prefixos

O `create-managed-prefix-list` exemplo a seguir cria uma lista de IPv4 prefixos com no máximo 10 entradas e cria 2 entradas na lista de prefixos.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [CreateManagedPrefixList](#) na Referência de AWS CLI Comandos.

create-nat-gateway

O código de exemplo a seguir mostra como usar create-nat-gateway.

AWS CLI

Exemplo 1: Para criar um NAT gateway público

O create-nat-gateway exemplo a seguir cria um NAT gateway público na sub-rede especificada e associa o endereço IP elástico ao ID de alocação especificado. Ao criar um NAT gateway público, você deve associar um endereço IP elástico.

```
aws ec2 create-nat-gateway \  
  --subnet-id subnet-0250c25a1fEXAMPLE \  
  --allocation-id eipalloc-09ad461b0dEXAMPLE
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

Exemplo 2: Para criar um NAT gateway privado

O `create-nat-gateway` exemplo a seguir cria um NAT gateway privado na sub-rede especificada. Um NAT gateway privado não tem um endereço IP elástico associado.

```
aws ec2 create-nat-gateway \  
  --subnet-id subnet-0250c25a1fEXAMPLE \  
  --connectivity-type private
```

Saída:

```
{  
  "NatGateway": {  
    "CreateTime": "2021-12-01T22:26:00.000Z",  
    "NatGatewayAddresses": [  
      {}  
    ],  
    "NatGatewayId": "nat-011b568379EXAMPLE",  
    "State": "pending",  
    "SubnetId": "subnet-0250c25a1fEXAMPLE",  
    "VpcId": "vpc-0a60eb65b4EXAMPLE",  
    "ConnectivityType": "private"  
  }  
}
```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [CreateNatGateway](#) na Referência de AWS CLI Comandos.

`create-network-acl-entry`

O código de exemplo a seguir mostra como usar `create-network-acl-entry`.

AWS CLI

Para criar uma ACL entrada de rede

Este exemplo cria uma entrada para a rede especificada ACL. A regra permite o tráfego de entrada de qualquer IPv4 endereço (0.0.0.0/0) na UDP porta 53 (DNS) em qualquer sub-rede associada. Se o comando for bem-sucedido, nenhuma saída será retornada.

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
```

Este exemplo cria uma regra para a rede especificada ACL que permite o tráfego de entrada de qualquer IPv6 endereço (:: /0) na TCP porta 80 (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
```

- Para API obter detalhes, consulte [CreateNetworkAclEntry](#) na Referência de AWS CLI Comandos.

create-network-acl

O código de exemplo a seguir mostra como usar create-network-acl.

AWS CLI

Para criar uma rede ACL

Este exemplo cria uma rede ACL para o especificado VPC.

Comando:

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

Saída:

```
{
  "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
}
}

```

- Para API obter detalhes, consulte [CreateNetworkAcl](#) na Referência de AWS CLI Comandos.

create-network-insights-access-scope

O código de exemplo a seguir mostra como usar `create-network-insights-access-scope`.

AWS CLI

Para criar um escopo de acesso à rede

O `create-network-insights-access-scope` exemplo a seguir cria um escopo de acesso à rede.

```

aws ec2 create-network-insights-access-scope \
  --cli-input-json file://access-scope-file.json

```

Conteúdo de `access-scope-file.json`:

```

{
  "MatchPaths": [
    {
      "Source": {
        "ResourceStatement": {
          "Resources": [

```

```

        "vpc-abcd12e3"
      ]
    }
  }
],
"ExcludePaths": [
  {
    "Source": {
      "ResourceStatement": {
        "ResourceTypes": [
          "AWS::EC2::InternetGateway"
        ]
      }
    }
  }
]
}

```

Saída:

```

{
  "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"
          ]
        }
      }
    ]
  }
}

```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI](#) Network Access Analyzer.

- Para API obter detalhes, consulte [CreateNetworkInsightsAccessScope](#) na Referência de AWS CLI Comandos.

create-network-insights-path

O código de exemplo a seguir mostra como usar `create-network-insights-path`.

AWS CLI

Para criar um caminho

O `create-network-insights-path` exemplo a seguir cria um caminho. A origem é o gateway de internet especificado e o destino é a EC2 instância especificada. Para determinar se o destino pode ser acessado usando o protocolo e a porta especificados, analise o caminho usando o `start-network-insights-analysis` comando.

```

aws ec2 create-network-insights-path \
  --source igw-0797cccdc9d73b0e5 \
  --destination i-0495d385ad28331c7 \
  --destination-port 22 \
  --protocol TCP

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Introdução ao uso do AWS CLI no Guia](#) do Reachability Analyzer.

- Para API obter detalhes, consulte [CreateNetworkInsightsPath](#) na Referência de AWS CLI Comandos.

create-network-interface-permission

O código de exemplo a seguir mostra como usar `create-network-interface-permission`.

AWS CLI

Para criar uma permissão de interface de rede

Este exemplo concede permissão à conta 123456789012 para vincular a interface de rede `eni-1a2b3c4d` a uma instância.

Comando:

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

Saída:

```

{
  "InterfacePermission": {
    "PermissionState": {
      "State": "GRANTED"
    },
    "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
    "NetworkInterfaceId": "eni-1a2b3c4d",
    "Permission": "INSTANCE-ATTACH",
  }
}

```

```
    "AwsAccountId": "123456789012"
  }
}
```

- Para API obter detalhes, consulte [CreateNetworkInterfacePermission](#) na Referência de AWS CLI Comandos.

create-network-interface

O código de exemplo a seguir mostra como usar `create-network-interface`.

AWS CLI

Exemplo 1: Para especificar um IPv4 endereço para uma interface de rede

O `create-network-interface` exemplo a seguir cria uma interface de rede para a sub-rede especificada com o IPv4 endereço primário especificado.

```
aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my network interface" \
  --groups sg-09dfba7ed20cda78b \
  --private-ip-address 10.0.8.17
```

Saída:

```
{
  "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"
  }
}

```

Exemplo 2: Para criar uma interface de rede com um IPv4 endereço e um IPv6 endereço

O `create-network-interface` exemplo a seguir cria uma interface de rede para a sub-rede especificada com um IPv4 endereço e um IPv6 endereço selecionados pela AmazonEC2.

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my dual stack network interface" \
  --ipv6-address-count 1 \
  --groups sg-09dfba7ed20cda78b

```

Saída:

```

{
  "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"
}
}

```

Exemplo 3: Para criar uma interface de rede com opções de configuração de rastreamento de conexão

O `create-network-interface` exemplo a seguir cria uma interface de rede e configura os tempos limite de rastreamento de conexão ociosa.

```

aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --groups sg-02e57dbcfe0331c1b \
  --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60

```

Saída:

```

{
  "NetworkInterface": {

```

```

    "AvailabilityZone": "us-west-2a",
    "ConnectionTrackingConfiguration": {
      "TcpEstablishedTimeout": 86400,
      "UdpTimeout": 60
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "my-security-group",
        "GroupId": "sg-02e57dbcfe0331c1b"
      }
    ],
    "InterfaceType": "interface",
    "Ipv6Addresses": [],
    "MacAddress": "06:4c:53:de:6d:91",
    "NetworkInterfaceId": "eni-0c133586e08903d0b",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
    "PrivateIpAddress": "10.0.8.94",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-8-94.us-west-2.compute.internal",
        "PrivateIpAddress": "10.0.8.94"
      }
    ],
    "RequesterId": "AIDA4Z3Y7GSXTMEXAMPLE",
    "RequesterManaged": false,
    "SourceDestCheck": true,
    "Status": "pending",
    "SubnetId": "subnet-00a24d0d67acf6333",
    "TagSet": [],
    "VpcId": "vpc-02723a0feeeb9d57b"
  }
}

```

Exemplo 4: Para criar um adaptador de malha elástica

O `create-network-interface` exemplo a seguir cria umEFA.

```

aws ec2 create-network-interface \
  --interface-type efa \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my efa" \

```

```
--groups sg-02e57dbcfe0331c1b
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Interfaces de rede elásticas](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [CreateNetworkInterface](#) na Referência de AWS CLI Comandos.

create-placement-group

O código de exemplo a seguir mostra como usar `create-placement-group`.

AWS CLI

Para criar um grupo de colocação

Esse exemplo de comando cria um grupo de posicionamento com o nome especificado.

Comando:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

Para criar um grupo de posicionamento de partições

Esse exemplo de comando cria um grupo de posicionamento de partições chamado HDFS-Group-A com cinco partições.

Comando:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

- Para API obter detalhes, consulte [CreatePlacementGroup](#) na Referência de AWS CLI Comandos.

create-replace-root-volume-task

O código de exemplo a seguir mostra como usar `create-replace-root-volume-task`.

AWS CLI

Exemplo 1: Para restaurar um volume raiz ao estado inicial de inicialização

O `create-replace-root-volume-task` exemplo a seguir restaura o volume raiz da instância `i-0123456789abcdefa` para seu estado inicial de inicialização.

```
aws ec2 create-replace-root-volume-task \  
  --instance-id i-0123456789abcdefa
```

Saída:

```
{
  "ReplaceRootVolumeTask":
  {
    "InstanceId": "i-0123456789abcdefa",
    "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
    "TaskState": "pending",
    "StartTime": "2022-03-14T15:06:38Z",
    "Tags": []
  }
}
```

Para obter mais informações, consulte [Substituir um volume raiz](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Exemplo 2: Para restaurar um volume raiz em um snapshot específico

O `create-replace-root-volume-task` exemplo a seguir restaura o volume raiz da instância `i-0123456789abcdefa` para o snapshot `snap-0abcdef1234567890`.

```
aws ec2 create-replace-root-volume-task \
  --instance-id i-0123456789abcdefa \
  --snapshot-id snap-0abcdef1234567890
```

Saída:

```
{
  "ReplaceRootVolumeTask":
  {
    "InstanceId": "i-0123456789abcdefa",
    "ReplaceRootVolumeTaskId": "replacevol-0555566667777abcd",
    "TaskState": "pending",
    "StartTime": "2022-03-14T15:16:28Z",
    "Tags": []
  }
}
```

Para obter mais informações, consulte [Substituir um volume raiz](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [CreateReplaceRootVolumeTask](#) na Referência de AWS CLI Comandos.

create-reserved-instances-listing

O código de exemplo a seguir mostra como usar `create-reserved-instances-listing`.

AWS CLI

Para listar uma instância reservada no Reserved Instance Marketplace

O `create-reserved-instances-listing` exemplo a seguir cria uma listagem para a Instância Reservada especificada no 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
```

- Para API obter detalhes, consulte [CreateReservedInstancesListing](#) na Referência de AWS CLI Comandos.

create-restore-image-task

O código de exemplo a seguir mostra como usar `create-restore-image-task`.

AWS CLI

Para restaurar e a AMI partir de um bucket do S3

O `create-restore-image-task` exemplo a seguir restaura e AMI de um bucket do S3. Use os valores `S3ObjectKey` e `Bucket` da `describe-store-image-tasks` saída, especifique a chave do objeto do AMI e o nome do bucket do S3 para o qual AMI foi copiado e especifique o nome do restaurado. AMI O nome dessa conta deve ser exclusivo AMIs na região. O restaurado AMI receberá um novo AMI ID.

```
aws ec2 create-restore-image-task \  
  --object-key ami-1234567890abcdef0.bin \  
  --bucket my-ami-bucket \  
  --name "New AMI Name"
```

Saída:

```
{
  "ImageId": "ami-0eab20fe36f83e1a8"
}
```

Para obter mais informações sobre como armazenar, restaurar e AMI usar o S3, consulte [Armazenar e restaurar e AMI usar o S3 <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuideami-store-restore.html>](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuideami-store-restore.html) no Guia do usuário da Amazon. EC2

- Para API obter detalhes, consulte [CreateRestoreImageTask](#) na Referência de AWS CLI Comandos.

create-route-table

O código de exemplo a seguir mostra como usar `create-route-table`.

AWS CLI

Para criar uma tabela de rotas

Este exemplo cria uma tabela de rotas para o especificado VPC.

Comando:

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

Saída:

```
{
  "RouteTable": {
    "Associations": [],
    "RouteTableId": "rtb-22574640",
    "VpcId": "vpc-a01106c2",
    "PropagatingVgws": [],
    "Tags": [],
    "Routes": [
      {
        "GatewayId": "local",
        "DestinationCidrBlock": "10.0.0.0/16",
        "State": "active"
      }
    ]
  }
}
```

```
}
```

- Para API obter detalhes, consulte [CreateRouteTable](#) na Referência de AWS CLI Comandos.

create-route

O código de exemplo a seguir mostra como usar create-route.

AWS CLI

Para criar uma rota

Este exemplo cria uma rota para a tabela de rotas especificada. A rota corresponde a todo o IPv4 tráfego (0.0.0.0/0) e o encaminha para o gateway de Internet especificado. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

Este exemplo de comando cria uma rota na tabela de rotas *rtb-g8ff4ea2*. A rota corresponde ao tráfego do IPv4 CIDR bloco 10.0.0.0/16 e o encaminha para a conexão de VPC peering, *pcx-111aaa22*. Essa rota permite que o tráfego seja direcionado para o par VPC na conexão de VPC peering. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

Este exemplo cria uma rota na tabela de rotas especificada que corresponde a todo o IPv6 tráfego (::/0) e a encaminha para o gateway de Internet somente de saída especificado.

Comando:

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-block ::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- Para API obter detalhes, consulte [CreateRoute](#) na Referência de AWS CLI Comandos.

create-security-group

O código de exemplo a seguir mostra como usar `create-security-group`.

AWS CLI

Para criar um grupo de segurança para EC2 -Classic

Este exemplo cria um grupo de segurança chamado `MySecurityGroup`.

Comando:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

Saída:

```
{
  "GroupId": "sg-903004f8"
}
```

Para criar um grupo de segurança para EC2 - VPC

Este exemplo cria um grupo de segurança com `MySecurityGroup` o nome especificadoVPC.

Comando:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group" --vpc-id vpc-1a2b3c4d
```

Saída:

```
{
  "GroupId": "sg-903004f8"
}
```

Para obter mais informações, consulte [Using Security Groups](#) no Guia do usuário da AWS Command Line Interface.

- Para API obter detalhes, consulte [CreateSecurityGroup](#) na Referência de AWS CLI Comandos.

create-snapshot

O código de exemplo a seguir mostra como usar `create-snapshot`.

AWS CLI

Para criar um instantâneo

Esse exemplo de comando cria um instantâneo do volume com um ID do volume `vol-1234567890abcdef0` e uma breve descrição para identificar o instantâneo.

Comando:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot"
```

Saída:

```
{
  "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"
}
```

Para criar um instantâneo com tags

Esse exemplo de comando cria um instantâneo e aplica duas tags: `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}]'
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [CreateSnapshot](#) na Referência de AWS CLI Comandos.

create-snapshots

O código de exemplo a seguir mostra como usar create-snapshots.

AWS CLI

Exemplo 1: Para criar um instantâneo de vários volumes

O create-snapshots exemplo a seguir cria instantâneos de todos os volumes anexados à instância especificada.

```
aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --description "This is snapshot of a volume from my-instance"
```

Saída:

```
{
```

```

"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"
  },
  ...
]
}

```

Exemplo 2: Para criar um instantâneo de vários volumes com tags do volume de origem

O `create-snapshots` exemplo a seguir cria instantâneos de todos os volumes anexados à instância especificada e copia as tags de cada volume para o instantâneo correspondente.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --copy-tags-from-source volume \
  --description "This is snapshot of a volume from my-instance"

```

Saída:

```
{
```

```

"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"
  }
  ...
]
}

```

Exemplo 3: Para criar um instantâneo de vários volumes que não inclua o volume raiz

O `create-snapshots` exemplo a seguir cria um instantâneo de todos os volumes anexados à instância especificada, exceto o volume raiz.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true

```

Consulte um exemplo de saída no exemplo 1.

Exemplo 4: Para criar um instantâneo de vários volumes e adicionar tags

O `create-snapshots` exemplo a seguir cria instantâneos de todos os volumes anexados à instância especificada e adiciona duas tags a cada instantâneo.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},
{Key=costcenter,Value=123}]'

```

Consulte um exemplo de saída no exemplo 1.

- Para API obter detalhes, consulte [CreateSnapshots](#) na Referência de AWS CLI Comandos.

create-spot-datafeed-subscription

O código de exemplo a seguir mostra como usar `create-spot-datafeed-subscription`.

AWS CLI

Para criar um feed de dados da Instância Spot

O `create-spot-datafeed-subscription` exemplo a seguir cria um feed de dados da Instância Spot.

```
aws ec2 create-spot-datafeed-subscription \  
  --bucket my-bucket \  
  --prefix spot-data-feed
```

Saída:

```
{  
  "SpotDatafeedSubscription": {  
    "Bucket": "my-bucket",  
    "OwnerId": "123456789012",  
    "Prefix": "spot-data-feed",  
    "State": "Active"  
  }  
}
```

O feed de dados é armazenado no bucket do Amazon S3 que você especificou. Os nomes dos arquivos desse feed de dados têm o seguinte formato.

```
my-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-HH.n.abcd1234.gz
```

Para obter mais informações, consulte o [feed de dados da instância spot](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [CreateSpotDatafeedSubscription](#) na Referência de AWS CLI Comandos.

create-store-image-task

O código de exemplo a seguir mostra como usar `create-store-image-task`.

AWS CLI

Para armazenar um AMI em um bucket do S3

O `create-store-image-task` exemplo a seguir armazena um AMI em um bucket do S3. Especifique o ID do AMI e o nome do bucket do S3 no qual armazenar o AMI

```
aws ec2 create-store-image-task \  
  --image-id ami-1234567890abcdef0 \  
  --bucket my-ami-bucket
```

Saída:

```
{  
  "ObjectKey": "ami-1234567890abcdef0.bin"  
}
```

Para obter mais informações, consulte [Armazenar e restaurar e AMI usar o S3](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [CreateStoreImageTask](#) na Referência de AWS CLI Comandos.

create-subnet-cidr-reservation

O código de exemplo a seguir mostra como usar `create-subnet-cidr-reservation`.

AWS CLI

Para criar uma reserva de sub-rede CIDR

O `create-subnet-cidr-reservation` exemplo a seguir cria uma CIDR reserva de sub-rede para a sub-rede e CIDR o intervalo especificados.

```
aws ec2 create-subnet-cidr-reservation \  
  --subnet-id subnet-03c51e2eEXAMPLE \  
  --reservation-type prefix \  
  --
```

```
--cidr 10.1.0.20/26
```

Saída:

```
{
  "SubnetCidrReservation": {
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",
    "Cidr": "10.1.0.16/28",
    "ReservationType": "prefix",
    "OwnerId": "123456789012"
  }
}
```

Para obter mais informações, consulte [CIDRReservas de sub-rede](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [CreateSubnetCidrReservation](#) na Referência de AWS CLI Comandos.

create-subnet

O código de exemplo a seguir mostra como usar create-subnet.

AWS CLI

Exemplo 1: Para criar uma sub-rede somente com um IPv4 CIDR bloco

O create-subnet exemplo a seguir cria uma sub-rede no bloco especificado VPC com o IPv4 CIDR bloco especificado.

```
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}]
```

Saída:

```
{
  "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"
  }
}

```

Exemplo 2: Para criar uma sub-rede com blocos IPv4 e IPv6 CIDR

O `create-subnet` exemplo a seguir cria uma sub-rede no especificado VPC com os IPv6 CIDR blocos IPv4 especificados.

```

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}]

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 3: Para criar uma sub-rede somente com um IPv6 CIDR bloco

O `create-subnet` exemplo a seguir cria uma sub-rede no bloco especificado VPC com o IPv6 CIDR bloco especificado.

```

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}]

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [VPCse sub-redes no Guia VPC](#) do usuário da Amazon.

- Para API obter detalhes, consulte [CreateSubnet](#) na Referência de AWS CLI Comandos.

create-tags

O código de exemplo a seguir mostra como usar create-tags.

AWS CLI

Exemplo 1: Para adicionar uma tag a um recurso

O create-tags exemplo a seguir adiciona a tag Stack=production à imagem especificada ou substitui uma tag existente por AMI onde está Stack a chave da tag.

```
aws ec2 create-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=production
```

Para obter mais informações, consulte [Este é o título do tópico](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: Para adicionar tags a vários recursos

O `create-tags` exemplo a seguir adiciona (ou substitui) duas tags para uma instância AMI e uma. Uma das tags tem uma chave (`webserver`), mas nenhum valor (o valor é definido como uma string vazia). A outra tag tem uma chave (`stack`) e um valor (`Production`).

```
aws ec2 create-tags \  
  --resources ami-1a2b3c4d i-1234567890abcdef0 \  
  --tags Key=webserver,Value= Key=stack,Value=Production
```

Para obter mais informações, consulte [Este é o título do tópico](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 3: Para adicionar tags contendo caracteres especiais

O exemplo `create-tags` a seguir adiciona a tag `[Group]=test` para uma instância. Os colchetes (`[` e `]`) são caracteres especiais e devem ser recuados. Os exemplos a seguir também usam o caractere de continuação de linha apropriado para cada ambiente.

Se você estiver usando o Windows, coloque o elemento com caracteres especiais entre aspas duplas (`"`) e preceda cada caractere de aspas duplas com uma barra invertida (`\`) da seguinte maneira:

```
aws ec2 create-tags ^  
  --resources i-1234567890abcdef0 ^  
  --tags Key="[Group]",Value=test
```

Se você estiver usando o Windows PowerShell, coloque o elemento no valor que tem caracteres especiais com aspas duplas (`"`), preceda cada caractere de aspas duplas com uma barra invertida (`\`) e, em seguida, coloque toda a estrutura de chave e valor entre aspas simples (`'`) da seguinte forma:

```
aws ec2 create-tags `
```

```
--resources i-1234567890abcdef0 \  
--tags 'Key="[Group]",Value=test'
```

Se você estiver usando Linux ou OS X, coloque o elemento com caracteres especiais entre aspas duplas (") e coloque toda a estrutura de chave e valor entre aspas simples (') da seguinte maneira:

```
aws ec2 create-tags \  
  --resources i-1234567890abcdef0 \  
  --tags 'Key="[Group]",Value=test'
```

Para obter mais informações, consulte [Este é o título do tópico](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [CreateTags](#) na Referência de AWS CLI Comandos.

create-traffic-mirror-filter-rule

O código de exemplo a seguir mostra como usar `create-traffic-mirror-filter-rule`.

AWS CLI

Para criar uma regra de filtro para tráfego de entrada TCP

O `create-traffic-mirror-filter-rule` exemplo a seguir cria uma regra que você pode usar para espelhar todo o TCP tráfego de entrada. Antes de executar esse comando, use `create-traffic-mirror-filter` para criar o 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
```

Saída:

```
{
```

```

    "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"
  }

```

Para obter mais informações, consulte [Criar um filtro de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [CreateTrafficMirrorFilterRule](#) na Referência de AWS CLI Comandos.

create-traffic-mirror-filter

O código de exemplo a seguir mostra como usar `create-traffic-mirror-filter`.

AWS CLI

Para criar um filtro de espelhos de tráfego

O `create-traffic-mirror-filter` exemplo a seguir cria um filtro Traffic Mirror. Depois de criar o filtro, use `create-traffic-mirror-filter-rule` para adicionar regras ao filtro.

```

aws ec2 create-traffic-mirror-filter \
  --description "TCP Filter"

```

Saída:

```

{
  "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",
  "TrafficMirrorFilter": {
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
    "Description": "TCP Filter",
    "EgressFilterRules": [],

```

```

    "IngressFilterRules": [],
    "Tags": [],
    "NetworkServices": []
  }
}

```

Para obter mais informações, consulte [Criar um filtro de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [CreateTrafficMirrorFilter](#) na Referência de AWS CLI Comandos.

create-traffic-mirror-session

O código de exemplo a seguir mostra como usar create-traffic-mirror-session.

AWS CLI

Para criar uma sessão de espelhamento de tráfego

O create-traffic-mirror-session comando a seguir cria sessões de espelhamento de tráfego para a origem e o destino especificados para 25 bytes do pacote.

```

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

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [Criar uma sessão de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [CreateTrafficMirrorSession](#) na Referência de AWS CLI Comandos.

create-traffic-mirror-target

O código de exemplo a seguir mostra como usar create-traffic-mirror-target.

AWS CLI

Para criar um alvo do Network Load Balancer Traffic Mirror

O create-traffic-mirror-target exemplo a seguir cria um alvo do 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

```

Saída:

```

{
  "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"
}

```



```
}
```

Para criar um alvo do Traffic Mirror de rede

O `create-traffic-mirror-target` exemplo a seguir cria um alvo do Traffic Mirror da interface de rede.

```
aws ec2 create-traffic-mirror-target --description "Destino da interface de rede" -- eni-eni-01f6f631e network-interface-id EXAMPLE
```

Saída:

```
{
  "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",
  "TrafficMirrorTarget": {
    "Description": "Network interface target",
    "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",
    "TrafficMirrorTargetId": "tmt-02dcdb2abEXAMPLE",
    "OwnerId": "111122223333",
    "Type": "network-interface",
    "Tags": []
  }
}
```

Para obter mais informações, consulte [Criar um alvo de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [CreateTrafficMirrorTarget](#) na Referência de AWS CLI Comandos.

create-transit-gateway-connect-peer

O código de exemplo a seguir mostra como usar `create-transit-gateway-connect-peer`.

AWS CLI

Para criar um Transit Gateway Connect peer

O `create-transit-gateway-connect-peer` exemplo a seguir cria um Connect peer.

```
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
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  }  
}
```

Para obter mais informações, consulte os [anexos do Transit Gateway Connect](#) e os [pares do Transit Gateway Connect](#) no Guia do Transit Gateways.

- Para API obter detalhes, consulte [CreateTransitGatewayConnectPeer](#) na Referência de AWS CLI Comandos.

create-transit-gateway-connect

O código de exemplo a seguir mostra como usar `create-transit-gateway-connect`.

AWS CLI

Para criar um anexo do Transit Gateway Connect

O `create-transit-gateway-connect` exemplo a seguir cria um anexo Connect, com o protocolo “gre”, para o anexo especificado.

```
aws ec2 create-transit-gateway-connect \
  --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
  --options "Protocol=gre"
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte os [anexos do Transit Gateway Connect e os pares do Transit Gateway Connect](#) no Guia do Transit Gateways.

- Para API obter detalhes, consulte [CreateTransitGatewayConnect](#) na Referência de AWS CLI Comandos.

create-transit-gateway-multicast-domain

O código de exemplo a seguir mostra como usar `create-transit-gateway-multicast-domain`.

AWS CLI

Exemplo 1: Para criar um domínio IGMP multicast

O `create-transit-gateway-multicast-domain` exemplo a seguir cria um domínio multicast para o gateway de trânsito especificado. Com as fontes estáticas desativadas, qualquer instância em sub-redes associadas ao domínio multicast pode enviar tráfego multicast. Se pelo menos um membro usar o IGMP protocolo, você deverá ativar o IGMPv2 suporte.

```
aws ec2 create-transit-gateway-multicast-domain \  
  --transit-gateway-id tgw-0bf0bffe2aEXAMPLE \  
  --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

Saída:

```
{  
  "TransitGatewayMulticastDomain": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",  
    "TransitGatewayId": "tgw-0bf0bffe2aEXAMPLE",  
    "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"  
  }  
}
```

Exemplo 2: Para criar um domínio multicast estático

O `create-transit-gateway-multicast-domain` exemplo a seguir cria um domínio multicast para o gateway de trânsito especificado. Com as fontes estáticas ativadas, você deve adicionar fontes estaticamente.

```
aws ec2 create-transit-gateway-multicast-domain \  
  --transit-gateway-id tgw-0bf0bffe2aEXAMPLE \  
  --options StaticSourcesSupport=enable,Igmpv2Support=enable
```

```
--options StaticSourcesSupport=enable,Igmpv2Support=disable
```

Saída:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
    "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"
  }
}
```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [CreateTransitGatewayMulticastDomain](#) na Referência de AWS CLI Comandos.

create-transit-gateway-peering-attachment

O código de exemplo a seguir mostra como usar `create-transit-gateway-peering-attachment`.

AWS CLI

Para criar um anexo de emparelhamento do Transit Gateway

O `create-transit-gateway-peering-attachment` exemplo a seguir cria uma solicitação de anexo de emparelhamento entre os dois gateways de trânsito especificados.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Transit Gateway Peering Attachments](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [CreateTransitGatewayPeeringAttachment](#) na Referência de AWS CLI Comandos.

create-transit-gateway-policy-table

O código de exemplo a seguir mostra como usar `create-transit-gateway-policy-table`.

AWS CLI

Para criar uma tabela de políticas de gateway de trânsito

O `create-transit-gateway-policy-table` exemplo a seguir cria uma tabela de políticas de gateway de trânsito para o gateway de trânsito especificado.

```
aws ec2 create-transit-gateway-policy-table \  

```

```
--transit-gateway-id tgw-067f8505c18f0bd6e
```

Saída:

```
{
  "TransitGatewayPolicyTable": {
    "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
    "TransitGatewayId": "tgw-067f8505c18f0bd6e",
    "State": "pending",
    "CreationTime": "2023-11-28T16:36:43+00:00"
  }
}
```

Para obter mais informações, consulte as [tabelas de políticas do Transit Gateway](#) no Guia do usuário do Transit Gateway.

- Para API obter detalhes, consulte [CreateTransitGatewayPolicyTable](#) na Referência de AWS CLI Comandos.

create-transit-gateway-prefix-list-reference

O código de exemplo a seguir mostra como usar `create-transit-gateway-prefix-list-reference`.

AWS CLI

Para criar uma referência a uma lista de prefixos

O `create-transit-gateway-prefix-list-reference` exemplo a seguir cria uma referência à lista de prefixos especificada na tabela de rotas do gateway de trânsito especificada.

```
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-aaaaaabbbbb1111
```

Saída:

```
{
  "TransitGatewayPrefixListReference": {
```

```

    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-1111112222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "pending",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aaaaaabbbbb11111",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}

```

Para obter mais informações, consulte as [referências da lista de prefixos](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [CreateTransitGatewayPrefixListReferencena](#) Referência de AWS CLI Comandos.

create-transit-gateway-route-table

O código de exemplo a seguir mostra como usar `create-transit-gateway-route-table`.

AWS CLI

Para criar uma tabela de rotas do Transit Gateway

O `create-transit-gateway-route-table` exemplo a seguir cria uma tabela de rotas para o gateway de trânsito especificado.

```

aws ec2 create-transit-gateway-route-table \
  --transit-gateway-id tgw-0262a0e521EXAMPLE

```

Saída:

```

{
  "TransitGatewayRouteTable": {
    "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "State": "pending",
    "DefaultAssociationRouteTable": false,
  }
}

```



```
    "DefaultPropagationRouteTable": false,  
    "CreationTime": "2019-07-10T19:01:46.000Z"  
  }  
}
```

Para obter mais informações, consulte [Criar uma tabela de rotas do Transit Gateways](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [CreateTransitGatewayRouteTable](#) na Referência de AWS CLI Comandos.

create-transit-gateway-route

O código de exemplo a seguir mostra como usar `create-transit-gateway-route`.

AWS CLI

Para criar uma rota de gateway de trânsito

O `create-transit-gateway-route` exemplo a seguir cria uma rota, com o destino especificado, para a tabela de rotas especificada.

```
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
```

Saída:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  
      {  
        "ResourceId": "vpc-0065acced4EXAMPLE",  
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
        "ResourceType": "vpc"  
      }  
    ],  
    "Type": "static",  
    "State": "active"  
  }  
}
```

```
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [CreateTransitGatewayRoute](#) na Referência de AWS CLI Comandos.

create-transit-gateway-vpc-attachment

O código de exemplo a seguir mostra como usar `create-transit-gateway-vpc-attachment`.

AWS CLI

Exemplo 1: Para associar um gateway de trânsito a um VPC

O `create-transit-gateway-vpc-attachment` exemplo a seguir cria um anexo de gateway de trânsito ao especificado VPC.

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-0262a0e521EXAMPLE \
  --vpc-id vpc-07e8ffd50f49335df \
  --subnet-id subnet-0752213d59EXAMPLE
```

Saída:

```
{
  "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"
    }
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Criar um anexo de gateway de trânsito a um VPC](#) no Guia de gateways de trânsito.

Exemplo 2: Para associar um gateway de trânsito a várias sub-redes em um VPC

O `create-transit-gateway-vpc-attachment` exemplo a seguir cria um anexo de gateway de trânsito às VPC sub-redes especificadas.

```
aws ec2 create-transit-gateway-vpc-attachment \  
  --transit-gateway-id tgw-02f776b1a7EXAMPLE \  
  --vpc-id vpc-3EXAMPLE \  
  --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Criar um anexo de gateway de trânsito a um VPC](#) no Guia de gateways de trânsito.

- Para API obter detalhes, consulte [CreateTransitGatewayVpcAttachment](#) na Referência de AWS CLI Comandos.

create-transit-gateway

O código de exemplo a seguir mostra como usar `create-transit-gateway`.

AWS CLI

Para criar um gateway de trânsito

O `create-transit-gateway` exemplo a seguir cria um gateway de trânsito.

```
aws ec2 create-transit-gateway \
  --description MyTGW \
  --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Criar um gateway de trânsito](#) no Guia de gateways de trânsito.

- Para API obter detalhes, consulte [CreateTransitGateway](#) na Referência de AWS CLI Comandos.

create-verified-access-endpoint

O código de exemplo a seguir mostra como usar `create-verified-access-endpoint`.

AWS CLI

Para criar um endpoint de acesso verificado

O `create-verified-access-endpoint` exemplo a seguir cria um endpoint de acesso verificado para o grupo de acesso verificado especificado. A interface de rede e o grupo de segurança especificados devem pertencer aos mesmos 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}]
```

Saída:

```
{
  "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"
        }
    ]
}
}

```

Para obter mais informações, consulte [Endpoints de acesso verificado](#) no Guia do usuário do AWS Verified Access.

- Para API obter detalhes, consulte [CreateVerifiedAccessEndpoint](#) na Referência de AWS CLI Comandos.

create-verified-access-group

O código de exemplo a seguir mostra como usar `create-verified-access-group`.

AWS CLI

Para criar um grupo de acesso verificado

O `create-verified-access-group` exemplo a seguir cria um grupo de acesso verificado para a instância de acesso verificado especificada.

```
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}]
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Grupos de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [CreateVerifiedAccessGroup](#) na Referência de AWS CLI Comandos.

create-verified-access-instance

O código de exemplo a seguir mostra como usar `create-verified-access-instance`.

AWS CLI

Para criar uma instância de acesso verificado

O `create-verified-access-instance` exemplo a seguir cria uma instância de acesso verificado com uma tag `Name`.

```
aws ec2 create-verified-access-instance \  
  --tag-specifications ResourceType=verified-access-  
instance,Tags=[{Key=Name,Value=my-va-instance}]
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Instâncias de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [CreateVerifiedAccessInstance](#) na Referência de AWS CLI Comandos.

create-verified-access-trust-provider

O código de exemplo a seguir mostra como usar `create-verified-access-trust-provider`.

AWS CLI

Para criar um provedor confiável de acesso verificado

O `create-verified-access-trust-provider` exemplo a seguir configura um provedor confiável de acesso verificado usando o 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}]
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Provedores confiáveis para acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [CreateVerifiedAccessTrustProvider](#) na Referência de AWS CLI Comandos.

create-volume

O código de exemplo a seguir mostra como usar `create-volume`.

AWS CLI

Para criar um volume vazio de uso geral SSD (gp2)

O `create-volume` exemplo a seguir cria um volume de uso geral SSD (gp2) de 80 GiB na zona de disponibilidade especificada. Observe que a região atual deve ser `us-east-1`, ou você pode adicionar o `--region` parâmetro para especificar a região para o comando.

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --availability-zone us-east-1a
```

Saída:

```
{  
  "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 você não especificar um tipo de volume, o tipo de volume padrão será gp2.

```
aws ec2 create-volume \  
  --size 80 \  
  --availability-zone us-east-1a
```

Exemplo 2: Para criar um volume provisionado IOPS SSD (io1) a partir de um snapshot

O create-volume exemplo a seguir cria um volume provisionado IOPS SSD (io1) com 1000 provisionados IOPS na zona de disponibilidade especificada usando o snapshot especificado.

```
aws ec2 create-volume \  
  --volume-type io1 \  
  --iops 1000 \  
  --snapshot-id snap-066877671789bd71b \  
  --availability-zone us-east-1a
```

Saída:

```
{  
  "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
}
```

Exemplo 3: Para criar um volume criptografado

O `create-volume` exemplo a seguir cria um volume criptografado usando o padrão CMK para EBS criptografia. Se a criptografia por padrão estiver desativada, você deverá especificar o `--encrypted` parâmetro da seguinte forma.

```
aws ec2 create-volume \
  --size 80 \
  --encrypted \
  --availability-zone us-east-1a
```

Saída:

```
{
  "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 a criptografia por padrão estiver ativada, o comando de exemplo a seguir cria um volume criptografado, mesmo sem o `--encrypted` parâmetro.

```
aws ec2 create-volume \
```

```
--size 80 \  
--availability-zone us-east-1a
```

Se você usar o `--kms-key-id` parâmetro para especificar um cliente gerenciado CMK, deverá especificar o `--encrypted` parâmetro mesmo se a criptografia, por padrão, estiver ativada.

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --encrypted \  
  --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \  
  --availability-zone us-east-1a
```

Exemplo 4: Para criar um volume com tags

O `create-volume` exemplo a seguir cria um volume e adiciona duas tags.

```
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}]'
```

- Para API obter detalhes, consulte [CreateVolume](#) na Referência de AWS CLI Comandos.

create-vpc-endpoint-connection-notification

O código de exemplo a seguir mostra como usar `create-vpc-endpoint-connection-notification`.

AWS CLI

Para criar uma notificação de conexão de endpoint

Este exemplo cria uma notificação para um serviço de endpoint específico que alerta você quando os endpoints da interface se conectam ao seu serviço e quando os endpoints são aceitos para o seu serviço.

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
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateVpcEndpointConnectionNotification](#) na Referência de AWS CLI Comandos.

create-vpc-endpoint-service-configuration

O código de exemplo a seguir mostra como usar `create-vpc-endpoint-service-configuration`.

AWS CLI

Exemplo 1: Para criar uma configuração de serviço de endpoint para um endpoint de interface

O `create-vpc-endpoint-service-configuration` exemplo a seguir cria uma configuração de serviço de VPC endpoint usando o Network Load Balancer `nlb-vpce`. Esse exemplo também especifica que as solicitações para se conectar ao serviço por meio de um endpoint de interface devem ser aceitas.

```
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

Saída:

```
{
  "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"
    ]
  }
}
```

Exemplo 2: Para criar uma configuração de serviço de endpoint para um endpoint do Gateway Load Balancer

O `create-vpc-endpoint-service-configuration` exemplo a seguir cria uma configuração de serviço de VPC endpoint usando o Gateway Load GWLBService Balancer. As solicitações para se conectar ao serviço por meio de um endpoint do Gateway Load Balancer são aceitas 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
```

Saída:

```
{
  "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/GWLBSvc/123123123123abcc"
    ]
  }
}
```

Para obter mais informações, consulte [serviços de VPC endpoint](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [CreateVpcEndpointServiceConfiguration](#) na Referência de AWS CLI Comandos.

create-vpc-endpoint

O código de exemplo a seguir mostra como usar `create-vpc-endpoint`.

AWS CLI

Exemplo 1: Para criar um endpoint de gateway

O `create-vpc-endpoint` exemplo a seguir cria um VPC endpoint de gateway entre o Amazon S3 VPC `vpc-1a2b3c4d` e o Amazon S3 na região `us-east-1` associa a `rtb-11aa22bb` tabela de rotas ao endpoint.

```
aws ec2 create-vpc-endpoint \
  --vpc-id vpc-1a2b3c4d \
```

```
--service-name com.amazonaws.us-east-1.s3 \  
--route-table-ids rtb-11aa22bb
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de um endpoint de gateway](#) no AWS PrivateLink Guia.

Exemplo 2: Para criar um endpoint de interface

O `create-vpc-endpoint` exemplo a seguir cria um VPC endpoint de interface entre o Amazon S3 VPC `vpc-1a2b3c4d` e o Amazon S3 na `us-east-1` região. O comando cria o endpoint na sub-rede `subnet-1a2b3c4d`, o associa ao grupo `sg-1a2b3c4d` de segurança e adiciona uma tag com uma chave de "Serviço" e um valor de "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}]
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Criação de um endpoint de interface](#) no Guia do usuário do. AWS PrivateLink

Exemplo 3: Para criar um endpoint do Gateway Load Balancer

O `create-vpc-endpoint` exemplo a seguir cria um endpoint do Gateway Load Balancer entre VPC `vpc-111122223333aabbcc` e um serviço configurado usando um 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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte os [endpoints do Gateway Load Balancer no Guia](#) do usuário do. AWS PrivateLink

- Para API obter detalhes, consulte [CreateVpcEndpoint](#) na Referência de AWS CLI Comandos.

create-vpc-peering-connection

O código de exemplo a seguir mostra como usar `create-vpc-peering-connection`.

AWS CLI

Para criar uma conexão de VPC emparelhamento entre seu VPCs

Este exemplo solicita uma conexão de emparelhamento entre seu VPCs `vpc-1a2b3c4d` e o `vpc-11122233`.

Comando:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233
```

Saída:

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Initiating Request to 444455556666",
      "Code": "initiating-request"
    },
    "Tags": [],
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-1a2b3c4d",
      "CidrBlock": "10.0.0.0/28"
    },
    "VpcPeeringConnectionId": "pcx-111aaa111",
    "ExpirationTime": "2014-04-02T16:13:36.000Z",
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-11122233"
    }
  }
}
```

Para criar uma conexão de VPC emparelhamento com uma VPC em outra conta

Este exemplo solicita uma conexão de emparelhamento entre sua VPC (`vpc-1a2b3c4d`) e uma (`vpc-11122233`) que pertence à conta 123456789012. VPC AWS

Comando:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012
```

Para criar uma conexão de VPC emparelhamento com um VPC em uma região diferente

Este exemplo solicita uma conexão de emparelhamento entre você VPC na região atual (vpc-1a2b3c4d) e uma (vpc-11122233) na sua conta na região. 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
```

Este exemplo solicita uma conexão de emparelhamento entre você VPC na região atual (vpc-1a2b3c4d) e uma VPC (vpc-11122233) que pertence à conta 123456789012 que está na região. 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
```

- Para API obter detalhes, consulte [CreateVpcPeeringConnection](#) na Referência de AWS CLI Comandos.

create-vpc

O código de exemplo a seguir mostra como usar create-vpc.

AWS CLI

Exemplo 1: Para criar um VPC

O create-vpc exemplo a seguir cria um VPC com o IPv4 CIDR bloco especificado e uma tag Name.

```
aws ec2 create-vpc \  
  --cidr-block 10.0.0.0/16 \  
  --tag-specifications ResourceType=vpc, Tags=[{Key=Name, Value=MyVpc}]
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Exemplo 2: Para criar um VPC com localização dedicada

O `create-vpc` exemplo a seguir cria um VPC com o IPv4 CIDR bloco especificado e a localização dedicada.

```
aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --instance-tenancy dedicated
```

Saída:

```
{
```

```

    "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
    }
  }
}

```

Exemplo 3: Para criar um VPC com um IPv6 CIDR bloco

O `create-vpc` exemplo a seguir cria um VPC com um bloco fornecido pela Amazon IPv6CIDR.

```

aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --amazon-provided-ipv6-cidr-block

```

Saída:

```

{
  "Vpc": {
    "CidrBlock": "10.0.0.0/16",
    "DhcpOptionsId": "dopt-dEXAMPLE",
    "State": "pending",
    "VpcId": "vpc-0fc5e3406bEXAMPLE",
    "OwnerId": "123456789012",
    "InstanceTenancy": "default",
    "Ipv6CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-068432c60bEXAMPLE",
        "Ipv6CidrBlock": "",

```

```

        "Ipv6CidrBlockState": {
            "State": "associating"
        },
        "Ipv6Pool": "Amazon",
        "NetworkBorderGroup": "us-west-2"
    }
],
"CidrBlockAssociationSet": [
    {
        "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
            "State": "associated"
        }
    }
],
"IsDefault": false
}
}

```

Exemplo 4: Para criar um VPC com a CIDR a partir de um IPAM pool

O `create-vpc` exemplo a seguir cria um VPC com CIDR a de um pool do 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"}]
```

Saída:

```
{
```

```
"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"
    }
  ]
}
}
```

Para obter mais informações, consulte [Criar um VPC que use um IPAM pool CIDR](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [CreateVpc](#) na Referência de AWS CLI Comandos.

create-vpn-connection-route

O código de exemplo a seguir mostra como usar `create-vpn-connection-route`.

AWS CLI

Para criar uma rota estática para uma VPN conexão

Este exemplo cria uma rota estática para a VPN conexão especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- Para API obter detalhes, consulte [CreateVpnConnectionRoute](#) na Referência de AWS CLI Comandos.

create-vpn-connection

O código de exemplo a seguir mostra como usar `create-vpn-connection`.

AWS CLI

Exemplo 1: Para criar uma VPN conexão com roteamento dinâmico

O `create-vpn-connection` exemplo a seguir cria uma VPN conexão entre o gateway privado virtual especificado e o gateway do cliente especificado e aplica tags à VPN conexão. A saída inclui as informações de configuração do dispositivo de gateway do cliente, em XML formato.

```
aws ec2 create-vpn-connection \  
  --type ipsec.1 \  
  --customer-gateway-id cgw-001122334455aabb \  
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b \  
  --tag-specification 'ResourceType=vpn-connection, Tags=[{Key=Name, Value=BGP-VPN}]'
```

Saída:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabb",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-123123123123abcab",  
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b",  
    "Options": {
```

```

    "EnableAcceleration": false,
    "StaticRoutesOnly": false,
    "LocalIpv4NetworkCidr": "0.0.0.0/0",
    "RemoteIpv4NetworkCidr": "0.0.0.0/0",
    "TunnelInsideIpVersion": "ipv4",
    "TunnelOptions": [
      {},
      {}
    ]
  },
  "Routes": [],
  "Tags": [
    {
      "Key": "Name",
      "Value": "BGP-VPN"
    }
  ]
}
}

```

Para obter mais informações, consulte [Como VPN funciona AWS de site a site no Guia do usuário de site](#) a AWS site VPN.

Exemplo 2: Para criar uma VPN conexão com roteamento estático

O `create-vpn-connection` exemplo a seguir cria uma VPN conexão entre o gateway privado virtual especificado e o gateway do cliente especificado. As opções especificam o roteamento estático. A saída inclui as informações de configuração do dispositivo de gateway do cliente, em XML formato.

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabbc \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --options "{\"StaticRoutesOnly\":true}"

```

Saída:

```

{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbc",

```

```

    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-123123123123abcb",
    "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": []
  }
}

```

Para obter mais informações, consulte [Como VPN funciona AWS de site a site no Guia do usuário de site](#) a AWS site VPN.

Exemplo 3: Para criar uma VPN conexão e especificar sua própria chave interna CIDR e pré-compartilhada

O `create-vpn-connection` exemplo a seguir cria uma VPN conexão e especifica o CIDR bloco de endereço IP interno e uma chave pré-compartilhada personalizada para cada túnel. Os valores especificados são retornados nas `CustomerGatewayConfiguration` informações.

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabb \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --options
  TunnelOptions='[{"TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},
{"TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'

```

Saída:

```

{
  "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": [
    {
      "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": []
}
}

```

Para obter mais informações, consulte [Como VPN funciona AWS de site a site no Guia do usuário de site](#) a AWS site VPN.

Exemplo 4: Para criar uma VPN conexão que ofereça suporte ao IPv6 tráfego

O `create-vpn-connection` exemplo a seguir cria uma VPN conexão que dá suporte ao IPv6 tráfego entre o gateway de trânsito especificado e o gateway do cliente especificado. As opções de túnel para ambos os túneis especificam que AWS deve iniciar a negociação. IKE

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --transit-gateway-id tgw-12312312312312312 \
  --customer-gateway-id cgw-001122334455aabbc \

```

```
--options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},  
{StartupAction=start}]
```

Saída:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "..configuration information...",  
    "CustomerGatewayId": "cgw-001122334455aabb",  
    "Category": "VPN",  
    "State": "pending",  
    "VpnConnectionId": "vpn-1111111122222222",  
    "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": []  
  }  
}
```

Para obter mais informações, consulte [Como VPN funciona AWS de site a site no Guia do usuário de site](#) a AWS site VPN.

- Para API obter detalhes, consulte [CreateVpnConnection](#) na Referência de AWS CLI Comandos.

create-vpn-gateway

O código de exemplo a seguir mostra como usar create-vpn-gateway.

AWS CLI

Para criar um gateway privado virtual

Este exemplo cria um gateway privado virtual.

Comando:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

Saída:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 64512,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

Para criar um gateway privado virtual com um lado específico da Amazon ASN

Este exemplo cria um gateway privado virtual e especifica o Número do Sistema Autônomo (ASN) para o lado Amazon da BGP sessão.

Comando:

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

Saída:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 65001,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

- Para API obter detalhes, consulte [CreateVpnGateway](#) na Referência de AWS CLI Comandos.

delete-carrier-gateway

O código de exemplo a seguir mostra como usar delete-carrier-gateway.

AWS CLI

Para excluir o gateway da operadora

O delete-carrier-gateway exemplo a seguir exclui o gateway da operadora especificado.

```
aws ec2 delete-carrier-gateway \
  --carrier-gateway-id cagw-0465cdEXAMPLE1111
```

Saída:

```
{
  "CarrierGateway": {
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
    "VpcId": "vpc-0c529aEXAMPLE1111",
    "State": "deleting",
    "OwnerId": "123456789012"
  }
}
```

Para obter mais informações, consulte [Gateways de operadora](#) no Guia do usuário da Amazon Virtual Private Cloud.

- Para API obter detalhes, consulte [DeleteCarrierGateway](#) na Referência de AWS CLI Comandos.

delete-client-vpn-endpoint

O código de exemplo a seguir mostra como usar delete-client-vpn-endpoint.

AWS CLI

Para excluir um VPN endpoint do cliente

O delete-client-vpn-endpoint exemplo a seguir exclui o VPN endpoint do cliente especificado.

```
aws ec2 delete-client-vpn-endpoint \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Saída:

```
{  
  "Status": {  
    "Code": "deleting"  
  }  
}
```

Para obter mais informações, consulte [Client VPN Endpoints](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [DeleteClientVpnEndpoint](#) na Referência de AWS CLI Comandos.

delete-client-vpn-route

O código de exemplo a seguir mostra como usar delete-client-vpn-route.

AWS CLI

Para excluir uma rota para um VPN endpoint do cliente

O delete-client-vpn-route exemplo a seguir exclui a 0.0.0.0/0 rota para a sub-rede especificada de um endpoint do ClienteVPN.

```
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
```

Saída:

```
{  
  "Status": {  
    "Code": "deleting"  
  }  
}
```


Para obter mais informações, consulte [Rotas](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [DeleteClientVpnRoute](#) na Referência de AWS CLI Comandos.

delete-coip-cidr

O código de exemplo a seguir mostra como usar `delete-coip-cidr`.

AWS CLI

Para excluir uma variedade de endereços IP de propriedade do cliente (CoIP)

O `delete-coip-cidr` exemplo a seguir exclui o intervalo especificado de endereços de CoIP no pool de CoIP especificado.

```
aws ec2 delete-coip-cidr \  
  --cidr 14.0.0.0/24 \  
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Saída:

```
{  
  "CoipCidr": {  
    "Cidr": "14.0.0.0/24",  
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"  
  }  
}
```

Para obter mais informações, consulte [Endereços IP pertencentes ao cliente](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [DeleteCoipCidr](#) na Referência de AWS CLI Comandos.

delete-coip-pool

O código de exemplo a seguir mostra como usar `delete-coip-pool`.

AWS CLI

Para excluir um pool de endereços IP de propriedade do cliente (CoIP)

O `delete-coip-pool` exemplo a seguir exclui um pool de endereços de CoIP.

```
aws ec2 delete-coip-pool \  
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Saída:

```
{  
  "CoipPool": {  
    "PoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-  
coip-1234567890abcdefg"  
  }  
}
```

Para obter mais informações, consulte [Endereços IP pertencentes ao cliente](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [DeleteCoipPool](#) na Referência de AWS CLI Comandos.

delete-customer-gateway

O código de exemplo a seguir mostra como usar `delete-customer-gateway`.

AWS CLI

Para excluir um gateway do cliente

Este exemplo exclui o gateway do cliente especificado. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- Para API obter detalhes, consulte [DeleteCustomerGateway](#) na Referência de AWS CLI Comandos.

delete-dhcp-options

O código de exemplo a seguir mostra como usar `delete-dhcp-options`.

AWS CLI

Para excluir um conjunto DHCP de opções

Este exemplo exclui o conjunto de DHCP opções especificado. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- Para API obter detalhes, consulte [DeleteDhcpOptions](#) na Referência de AWS CLI Comandos.

delete-egress-only-internet-gateway

O código de exemplo a seguir mostra como usar `delete-egress-only-internet-gateway`.

AWS CLI

Para excluir um gateway de Internet somente de saída

Este exemplo exclui o gateway de Internet somente de saída especificado.

Comando:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

Saída:

```
{  
  "ReturnCode": true  
}
```

- Para API obter detalhes, consulte [DeleteEgressOnlyInternetGateway](#) na Referência de AWS CLI Comandos.

delete-fleets

O código de exemplo a seguir mostra como usar `delete-fleets`.

AWS CLI

Exemplo 1: Para excluir uma EC2 frota e encerrar as instâncias associadas

O `delete-fleets` exemplo a seguir exclui a EC2 frota especificada e encerra as instâncias sob demanda e as instâncias spot associadas.

```
aws ec2 delete-fleets \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --terminate-instances
```

Saída:

```
{  
  "SuccessfulFleetDeletions": [  
    {  
      "CurrentFleetState": "deleted_terminating",  
      "PreviousFleetState": "active",  
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
    }  
  ],  
  "UnsuccessfulFleetDeletions": []  
}
```

Para obter mais informações, consulte [Excluir uma EC2 frota](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: excluir uma EC2 frota sem encerrar as instâncias associadas

O `delete-fleets` exemplo a seguir exclui a EC2 frota especificada sem encerrar as instâncias sob demanda e as instâncias spot associadas.

```
aws ec2 delete-fleets \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --no-terminate-instances
```

Saída:

```
{  
  "SuccessfulFleetDeletions": [  
    {
```

```
        "CurrentFleetState": "deleted_running",
        "PreviousFleetState": "active",
        "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
    }
],
"UnsuccessfulFleetDeletions": []
}
```

Para obter mais informações, consulte [Excluir uma EC2 frota](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DeleteFleets](#) na Referência de AWS CLI Comandos.

delete-flow-logs

O código de exemplo a seguir mostra como usar `delete-flow-logs`.

AWS CLI

Para excluir um registro de fluxo

O `delete-flow-logs` exemplo a seguir exclui o registro de fluxo especificado.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

Saída:

```
{
  "Unsuccessful": []
}
```

- Para API obter detalhes, consulte [DeleteFlowLogs](#) na Referência de AWS CLI Comandos.

delete-fpga-image

O código de exemplo a seguir mostra como usar `delete-fpga-image`.

AWS CLI

Para excluir uma FPGA imagem da Amazon

Este exemplo exclui o especificadoAFI.

Comando:

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [DeleteFpgaImagem](#) na Referência de AWS CLI Comandos.

delete-instance-connect-endpoint

O código de exemplo a seguir mostra como usar delete-instance-connect-endpoint.

AWS CLI

Para excluir um endpoint do EC2 Instance Connect

O delete-instance-connect-endpoint exemplo a seguir exclui o Endpoint do EC2 Instance Connect especificado.

```
aws ec2 delete-instance-connect-endpoint \
  --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

Saída:

```
{
  "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"
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Remove EC2 Instance Connect Endpoint](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DeleteInstanceConnectEndpoint](#) na Referência de AWS CLI Comandos.

delete-instance-event-window

O código de exemplo a seguir mostra como usar `delete-instance-event-window`.

AWS CLI

Exemplo 1: Para excluir uma janela de evento

O `delete-instance-event-window` exemplo a seguir exclui uma janela de evento.

```
aws ec2 delete-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890
```

Saída:

```
{  
  "InstanceEventWindowState": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "State": "deleting"  
  }  
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 2: Para forçar a exclusão de uma janela de eventos

O `delete-instance-event-window` exemplo a seguir exclui à força uma janela de eventos se a janela de eventos estiver atualmente associada a alvos.

```
aws ec2 delete-instance-event-window \  
  --force
```

```
--region us-east-1 \  
--instance-event-window-id iew-0abcdef1234567890 \  
--force-delete
```

Saída:

```
{  
  "InstanceEventWindowState": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "State": "deleting"  
  }  
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [DeleteInstanceEventWindow](#) na Referência de AWS CLI Comandos.

delete-internet-gateway

O código de exemplo a seguir mostra como usar delete-internet-gateway.

AWS CLI

Para excluir um gateway de internet

O delete-internet-gateway exemplo a seguir exclui o gateway de internet especificado.

```
aws ec2 delete-internet-gateway \  
--internet-gateway-id igw-0d0fb496b3EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Gateways de Internet](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DeleteInternetGateway](#) na Referência de AWS CLI Comandos.

delete-ipam-pool

O código de exemplo a seguir mostra como usar delete-ipam-pool.

AWS CLI

Para excluir um IPAM pool

Neste exemplo, você é um administrador IPAM delegado que deseja excluir um IPAM pool de que não precisa mais, mas o pool tem um CIDR provisionado para ele. Você não pode excluir um pool se ele tiver sido CIDRs provisionado a ele, a menos que use a `--cascade` opção, então você usará. `--cascade`

Para concluir essa solicitação:

Você precisará do ID do IPAM pool que você pode obter com [describe-ipam-pools](#). `--region` Deve ser a região de IPAM origem.

O `delete-ipam-pool` exemplo a seguir exclui um IPAM pool na sua AWS conta.

```
aws ec2 delete-ipam-pool \  
  --ipam-pool-id ipam-pool-050c886a3ca41cd5b \  
  --cascade \  
  --region us-east-1
```

Saída:

```
{  
  "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
```

```
}
}
```

Para obter mais informações, consulte [Excluir um pool](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DeletelpamPool](#) na Referência de AWS CLI Comandos.

delete-ipam-resource-discovery

O código de exemplo a seguir mostra como usar `delete-ipam-resource-discovery`.

AWS CLI

Para excluir uma descoberta de recursos

Neste exemplo, você é um administrador IPAM delegado que deseja excluir uma descoberta de recursos não padrão que você criou para compartilhar com outro IPAM administrador durante o processo de integração IPAM com contas fora da sua organização.

Para concluir essa solicitação:

`--region` Deve ser a região em que você criou a descoberta do recurso. Você não pode excluir uma descoberta de recurso padrão se. `"IsDefault": true` Uma descoberta de recursos padrão é aquela que é criada automaticamente na conta que cria uma IPAM. Para excluir uma descoberta de recursos padrão, você precisa excluir IPAM o.

O `delete-ipam-resource-discovery` exemplo a seguir exclui uma descoberta de recursos.

```
aws ec2 delete-ipam-resource-discovery \
  --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \
  --region us-east-1
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações sobre descobertas de recursos, consulte [Trabalhar com descobertas de recursos no Guia VPC](#) IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DeletelpamResourceDiscovery](#) na Referência de AWS CLI Comandos.

delete-ipam-scope

O código de exemplo a seguir mostra como usar delete-ipam-scope.

AWS CLI

Para excluir um IPAM escopo

O delete-ipam-scope exemplo a seguir exclui um IPAM.

```

aws ec2 delete-ipam-scope \
  --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4

```

Saída:

```

{
  "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"
  }
}
```

Para obter mais informações, consulte [Excluir um escopo](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DeletelpamScope](#) na Referência de AWS CLI Comandos.

delete-ipam

O código de exemplo a seguir mostra como usar `delete-ipam`.

AWS CLI

Para excluir um IPAM

O `delete-ipam` exemplo a seguir exclui um IPAM.

```
aws ec2 delete-ipam \
  --ipam-id ipam-036486dfa6af58ee0
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Excluir um IPAM](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteIpam](#) na Referência de AWS CLI Comandos.

delete-key-pair

O código de exemplo a seguir mostra como usar `delete-key-pair`.

AWS CLI

Para excluir um par de chaves

O `delete-key-pair` exemplo a seguir exclui o par de chaves especificado.

```
aws ec2 delete-key-pair \
  --key-name my-key-pair
```

Saída:

```
{
  "Return": true,
  "KeyPairId": "key-03c8d3aceb53b507"
}
```

Para obter mais informações, consulte [Criar e excluir pares de chaves](#) no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DeleteKeyPair](#) na Referência de AWS CLI Comandos.

delete-launch-template-versions

O código de exemplo a seguir mostra como usar `delete-launch-template-versions`.

AWS CLI

Para excluir uma versão do modelo de lançamento

Este exemplo exclui a versão do modelo de lançamento especificado.

Comando:

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --  
versions 1
```

Saída:

```
{  
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],  
  "SuccessfullyDeletedLaunchTemplateVersions": [  
    {  
      "LaunchTemplateName": "TestVersion",  
      "VersionNumber": 1,  
      "LaunchTemplateId": "lt-0abcd290751193123"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DeleteLaunchTemplateVersions](#) na Referência de AWS CLI Comandos.

delete-launch-template

O código de exemplo a seguir mostra como usar delete-launch-template.

AWS CLI

Para excluir um modelo de execução

Este exemplo exclui o modelo de execução especificado.

Comando:

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

Saída:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "TestTemplate",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- Para API obter detalhes, consulte [DeleteLaunchTemplate](#) na Referência de AWS CLI Comandos.

delete-local-gateway-route-table-virtual-interface-group-association

O código de exemplo a seguir mostra como usar `delete-local-gateway-route-table-virtual-interface-group-association`.

AWS CLI

Para desassociar uma tabela de rotas de gateway local de um grupo de interfaces virtuais (VIFs)

O `delete-local-gateway-route-table-virtual-interface-group-association` exemplo a seguir exclui a associação entre a tabela e o VIF grupo de rotas do gateway local especificado.

```
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
```

Saída:

```
{
  "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": []
  }
}

```

Para obter mais informações, consulte [associações de VIF grupos](#) no Guia do Usuário do AWS Outposts.

- Para API obter detalhes, consulte [DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#) na Referência de AWS CLI Comandos.

delete-local-gateway-route-table-vpc-association

O código de exemplo a seguir mostra como usar `delete-local-gateway-route-table-vpc-association`.

AWS CLI

Para desassociar uma tabela de rotas de gateway local de um VPC

O `delete-local-gateway-route-table-vpc-association` exemplo a seguir exclui a associação entre a tabela de rotas do gateway local especificada e VPC

```

aws ec2 delete-local-gateway-route-table-vpc-association \
  --local-gateway-route-table-vpc-association-id vpc-example0123456789

```

Saída:

```

{
  "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"
  }
}
```

Para obter mais informações, consulte [VPCassociações](#) no Guia do Usuário do AWS Outposts.

- Para API obter detalhes, consulte [DeleteLocalGatewayRouteTableVpcAssociation](#) na Referência de AWS CLI Comandos.

delete-local-gateway-route-table

O código de exemplo a seguir mostra como usar `delete-local-gateway-route-table`.

AWS CLI

Para excluir uma tabela de rotas de gateway local

O `delete-local-gateway-route-table` exemplo a seguir cria uma tabela de rotas de gateway local com o modo de VPC roteamento direto.

```
aws ec2 delete-local-gateway-route-table \
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Saída:

```
{
  "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"
  }
}
```

[Para obter mais informações, consulte Tabelas de rotas do gateway local](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [DeleteLocalGatewayRouteTable](#) na Referência de AWS CLI Comandos.

delete-local-gateway-route

O código de exemplo a seguir mostra como usar delete-local-gateway-route.

AWS CLI

Para excluir uma rota de uma tabela de rotas do gateway local

O delete-local-gateway-route exemplo a seguir exclui a rota especificada da tabela de rotas do gateway local especificada.

```
aws ec2 delete-local-gateway-route \
  --destination-cidr-block 0.0.0.0/0 \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

Saída:

```
{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
  }
}
```

- Para API obter detalhes, consulte [DeleteLocalGatewayRoute](#) na Referência de AWS CLI Comandos.

delete-managed-prefix-list

O código de exemplo a seguir mostra como usar delete-managed-prefix-list.

AWS CLI

Para excluir uma lista de prefixos

O `delete-managed-prefix-list` exemplo a seguir exclui a lista de prefixos especificada.

```
aws ec2 delete-managed-prefix-list \  
  --prefix-list-id pl-0123456abcabc1
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DeleteManagedPrefixList](#) na Referência de AWS CLI Comandos.

delete-nat-gateway

O código de exemplo a seguir mostra como usar `delete-nat-gateway`.

AWS CLI

Para excluir um NAT gateway

Este exemplo exclui o NAT gateway `nat-04ae55e711cec5680`.

Comando:

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

Saída:

```
{
  "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- Para API obter detalhes, consulte [DeleteNatGateway](#) na Referência de AWS CLI Comandos.

delete-network-acl-entry

O código de exemplo a seguir mostra como usar `delete-network-acl-entry`.

AWS CLI

Para excluir uma ACL entrada de rede

Este exemplo exclui a regra de entrada número 100 da rede especificada. ACL Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- Para API obter detalhes, consulte [DeleteNetworkAclEntry](#) na Referência de AWS CLI Comandos.

delete-network-acl

O código de exemplo a seguir mostra como usar `delete-network-acl`.

AWS CLI

Para excluir uma rede ACL

Este exemplo exclui a rede ACL especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- Para API obter detalhes, consulte [DeleteNetworkAcl](#) na Referência de AWS CLI Comandos.

delete-network-insights-access-scope-analysis

O código de exemplo a seguir mostra como usar `delete-network-insights-access-scope-analysis`.

AWS CLI

Para excluir uma análise do escopo de acesso à rede

O `delete-network-insights-access-scope-analysis` exemplo a seguir exclui a análise do escopo de acesso à rede especificada.

```
aws ec2 delete-network-insights-access-scope-analysis \  
  --network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

Saída:

```
{  
  "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef"  
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI Network Access Analyzer](#).

- Para API obter detalhes, consulte [DeleteNetworkInsightsAccessScopeAnalysis](#) na Referência de AWS CLI Comandos.

delete-network-insights-access-scope

O código de exemplo a seguir mostra como usar `delete-network-insights-access-scope`.

AWS CLI

Para excluir um escopo de acesso à rede

O `delete-network-insights-access-scope` exemplo a seguir exclui o escopo de acesso à rede especificado.

```
aws ec2 delete-network-insights-access-scope \  
--network-insights-access-scope-id nis-123456789abc01234
```

Saída:

```
{  
  "NetworkInsightsAccessScopeId": "nis-123456789abc01234"  
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI](#) Network Access Analyzer.

- Para API obter detalhes, consulte [DeleteNetworkInsightsAccessScope](#) na Referência de AWS CLI Comandos.

delete-network-insights-analysis

O código de exemplo a seguir mostra como usar delete-network-insights-analysis.

AWS CLI

Para excluir uma análise de caminho

O delete-network-insights-analysis exemplo a seguir exclui a análise especificada.

```
aws ec2 delete-network-insights-analysis \  
--network-insights-analysis-id nia-02207aa13eb480c7a
```

Saída:

```
{  
  "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"  
}
```

Para obter mais informações, consulte [Introdução ao uso do AWS CLI no Guia](#) do Reachability Analyzer.

- Para API obter detalhes, consulte [DeleteNetworkInsightsAnalysis](#) na Referência de AWS CLI Comandos.

delete-network-insights-path

O código de exemplo a seguir mostra como usar `delete-network-insights-path`.

AWS CLI

Para excluir um caminho

O `delete-network-insights-path` exemplo a seguir exclui o caminho especificado. Antes de excluir um caminho, você deve excluir todas as suas análises usando o `delete-network-insights-analysis` comando.

```
aws ec2 delete-network-insights-path \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

Saída:

```
{  
  "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"  
}
```

Para obter mais informações, consulte [Introdução ao uso do AWS CLI no Guia](#) do Reachability Analyzer.

- Para API obter detalhes, consulte [DeleteNetworkInsightsPath](#) na Referência de AWS CLI Comandos.

delete-network-interface-permission

O código de exemplo a seguir mostra como usar `delete-network-interface-permission`.

AWS CLI

Para excluir uma permissão de interface de rede

Este exemplo exclui a permissão de interface de rede especificada.

Comando:

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-  
perm-06fd19020ede149ea
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [DeleteNetworkInterfacePermission](#) na Referência de AWS CLI Comandos.

delete-network-interface

O código de exemplo a seguir mostra como usar `delete-network-interface`.

AWS CLI

Para excluir uma interface de rede

Este exemplo exclui a interface de rede especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- Para API obter detalhes, consulte [DeleteNetworkInterface](#) na Referência de AWS CLI Comandos.

delete-placement-group

O código de exemplo a seguir mostra como usar `delete-placement-group`.

AWS CLI

Para excluir um grupo de posicionamento

Esse exemplo de comando exclui o grupo de posicionamento especificado.

Comando:

```
aws ec2 delete-placement-group --group-name my-cluster
```


- Para API obter detalhes, consulte [DeletePlacementGroup](#) na Referência de AWS CLI Comandos.

delete-queued-reserved-instances

O código de exemplo a seguir mostra como usar `delete-queued-reserved-instances`.

AWS CLI

Para excluir uma compra em fila

O `delete-queued-reserved-instances` exemplo a seguir exclui a instância reservada especificada, que estava na fila para compra.

```
aws ec2 delete-queued-reserved-instances \  
  --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

Saída:

```
{  
  "SuccessfulQueuedPurchaseDeletions": [  
    {  
      "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"  
    }  
  ],  
  "FailedQueuedPurchaseDeletions": []  
}
```

- Para API obter detalhes, consulte [DeleteQueuedReservedInstances](#) na Referência de AWS CLI Comandos.

delete-route-table

O código de exemplo a seguir mostra como usar `delete-route-table`.

AWS CLI

Para excluir uma tabela de rotas

Este exemplo exclui a tabela de rotas especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- Para API obter detalhes, consulte [DeleteRouteTable](#) na Referência de AWS CLI Comandos.

delete-route

O código de exemplo a seguir mostra como usar `delete-route`.

AWS CLI

Para excluir uma rota

Este exemplo exclui a rota especificada da tabela de rotas especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0
```

- Para API obter detalhes, consulte [DeleteRoute](#) na Referência de AWS CLI Comandos.

delete-security-group

O código de exemplo a seguir mostra como usar `delete-security-group`.

AWS CLI

[EC2-Clássico] Para excluir um grupo de segurança

Este exemplo exclui o grupo de segurança chamado `MySecurityGroup`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] Para excluir um grupo de segurança

Este exemplo exclui o grupo de segurança com o ID `sg-903004f8`. Observe que você não pode referenciar um grupo de segurança para EC2 - VPC por nome. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-security-group --group-id sg-903004f8
```

Para obter mais informações, consulte [Using Security Groups](#) no Guia do usuário da AWS Command Line Interface.

- Para API obter detalhes, consulte [DeleteSecurityGroup](#) na Referência de AWS CLI Comandos.

delete-snapshot

O código de exemplo a seguir mostra como usar `delete-snapshot`.

AWS CLI

Para excluir um snapshot

Este exemplo de comando exclui um snapshot com o ID de snapshot de `snap-1234567890abcdef0`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- Para API obter detalhes, consulte [DeleteSnapshot](#) na Referência de AWS CLI Comandos.

delete-spot-datafeed-subscription

O código de exemplo a seguir mostra como usar `delete-spot-datafeed-subscription`.

AWS CLI

Para cancelar uma assinatura de feed de dados da Spot Instance

Este exemplo de comando exclui uma assinatura de feed de dados Spot para a conta. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-spot-datafeed-subscription
```

- Para API obter detalhes, consulte [DeleteSpotDatafeedSubscription](#) na Referência de AWS CLI Comandos.

delete-subnet-cidr-reservation

O código de exemplo a seguir mostra como usar `delete-subnet-cidr-reservation`.

AWS CLI

Para excluir uma reserva de sub-rede CIDR

O `delete-subnet-cidr-reservation` exemplo a seguir exclui a reserva de sub-rede CIDR especificada.

```
aws ec2 delete-subnet-cidr-reservation \  
  --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

Saída:

```
{  
  "DeletedSubnetCidrReservation": {  
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",  
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",  
    "Cidr": "10.1.0.16/28",  
    "ReservationType": "prefix",  
    "OwnerId": "123456789012"  
  }  
}
```

Para obter mais informações, consulte [CIDRReservas de sub-rede](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DeleteSubnetCidrReservation](#) na Referência de AWS CLI Comandos.

delete-subnet

O código de exemplo a seguir mostra como usar `delete-subnet`.

AWS CLI

Para excluir uma sub-rede

Este exemplo exclui a sub-rede especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- Para API obter detalhes, consulte [DeleteSubnet](#) na Referência de AWS CLI Comandos.

delete-tags

O código de exemplo a seguir mostra como usar delete-tags.

AWS CLI

Exemplo 1: Para excluir uma tag de um recurso

O delete-tags exemplo a seguir exclui a tag Stack=Test da imagem especificada. Quando você especifica um valor e um nome de chave, a tag é excluída somente se o valor da tag corresponder ao valor especificado.

```
aws ec2 delete-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=Test
```

É opcional especificar o valor de uma tag. O delete-tags exemplo a seguir exclui a tag com o nome purpose da chave da instância especificada, independentemente do valor da tag.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=purpose
```

Se você especificar a string vazia como o valor da tag, a tag será excluída somente se o valor da tag for a string vazia. O delete-tags exemplo a seguir especifica a string vazia como o valor da tag a ser excluída.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=Name,Value=
```

Exemplo 2: Para excluir uma tag de vários recursos

O delete-tags exemplo a seguir exclui a tag ``purpose=test`` de uma instância e de uma AMI. Conforme mostrado no exemplo anterior, você pode omitir o valor da tag do comando.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 ami-1234567890abcdef0 \  
  --tags Key=Purpose
```

- Para API obter detalhes, consulte [DeleteTags](#) na Referência de AWS CLI Comandos.

delete-traffic-mirror-filter-rule

O código de exemplo a seguir mostra como usar delete-traffic-mirror-filter-rule.

AWS CLI

Para excluir uma regra de filtro de espelhos de tráfego

O delete-traffic-mirror-filter-rule exemplo a seguir exclui a regra de filtro de espelhos de tráfego especificada.

```
aws ec2 delete-traffic-mirror-filter-rule \  
  --traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

Saída:

```
{  
  "TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"  
}
```

Para obter mais informações, consulte [Modificar suas regras de filtro de espelhos](#) de AWS tráfego no Guia de espelhamento de tráfego.

- Para API obter detalhes, consulte [DeleteTrafficMirrorFilterRule](#) na Referência de AWS CLI Comandos.

delete-traffic-mirror-filter

O código de exemplo a seguir mostra como usar `delete-traffic-mirror-filter`.

AWS CLI

Para excluir um filtro de espelhos de trânsito

O `delete-traffic-mirror-filter` exemplo a seguir exclui o filtro de espelhamento de tráfego especificado.

```
aws ec2 delete-traffic-mirror-filter \  
  --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

Saída:

```
{  
  "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"  
}
```

Para obter mais informações, consulte [Excluir um filtro de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [DeleteTrafficMirrorFilter](#) na Referência de AWS CLI Comandos.

delete-traffic-mirror-session

O código de exemplo a seguir mostra como usar `delete-traffic-mirror-session`.

AWS CLI

Para excluir uma sessão de espelhamento de tráfego

O `delete-traffic-mirror-session` exemplo a seguir exclui a sessão de espelhamento de tráfego especificada.

```
aws ec2 delete-traffic-mirror-session \  
  --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

Saída:

```
{
  "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"
}
```

Para obter mais informações, consulte [Excluir uma sessão de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [DeleteTrafficMirrorSession](#) na Referência de AWS CLI Comandos.

delete-traffic-mirror-target

O código de exemplo a seguir mostra como usar `delete-traffic-mirror-target`.

AWS CLI

Para excluir um alvo de espelhos de trânsito

O `delete-traffic-mirror-target` exemplo a seguir exclui o alvo do espelhamento de tráfego especificado.

```
aws ec2 delete-traffic-mirror-target \
  --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

Saída:

```
{
  "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"
}
```

Para obter mais informações, consulte [Excluir um alvo de espelhamento de tráfego](#) no Guia de espelhamento de AWS tráfego.

- Para API obter detalhes, consulte [DeleteTrafficMirrorTarget](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-connect-peer

O código de exemplo a seguir mostra como usar `delete-transit-gateway-connect-peer`.

AWS CLI

Para excluir um peer do Transit Gateway Connect

O `delete-transit-gateway-connect-peer` exemplo a seguir exclui o Connect peer especificado.

```
aws ec2 delete-transit-gateway-connect-peer \
  --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

Saída:

```
{
  "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"
        }
      ]
    }
  }
}
```

```
}
```

Para obter mais informações, consulte os [anexos do Transit Gateway Connect e os pares do Transit Gateway Connect](#) no Guia do Transit Gateways.

- Para API obter detalhes, consulte [DeleteTransitGatewayConnect](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-connect

O código de exemplo a seguir mostra como usar delete-transit-gateway-connect.

AWS CLI

Para excluir um anexo do Transit Gateway Connect

O delete-transit-gateway-connect exemplo a seguir exclui o anexo Connect especificado.

```
aws ec2 delete-transit-gateway-connect \
  --transit-gateway-attachment-id tgw-attach-037012e5dcEXAMPLE
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte os [anexos do Transit Gateway Connect e os pares do Transit Gateway Connect](#) no Guia do Transit Gateways.

- Para API obter detalhes, consulte [DeleteTransitGatewayConnect](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-multicast-domain

O código de exemplo a seguir mostra como usar `delete-transit-gateway-multicast-domain`.

AWS CLI

Para excluir um domínio multicast do Transit Gateway

O `delete-transit-gateway-multicast-domain` exemplo a seguir exclui o domínio multicast especificado.

```
aws ec2 delete-transit-gateway-multicast-domain \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Saída:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
    "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
    "State": "deleting",
    "CreationTime": "2019-11-20T22:02:03.000Z"
  }
}
```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DeleteTransitGatewayMulticastDomain](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-peering-attachment

O código de exemplo a seguir mostra como usar `delete-transit-gateway-peering-attachment`.

AWS CLI

Para excluir um anexo de emparelhamento do Transit Gateway

O `delete-transit-gateway-peering-attachment` exemplo a seguir exclui o anexo de emparelhamento do Transit Gateway especificado.

```
aws ec2 delete-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Transit Gateway Peering Attachments](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DeleteTransitGatewayPeeringAttachment](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-policy-table

O código de exemplo a seguir mostra como usar `delete-transit-gateway-policy-table`.

AWS CLI

Para excluir uma tabela de políticas do Transit Gateway

O `delete-transit-gateway-policy-table` exemplo a seguir exclui a tabela de políticas do gateway de trânsito especificada.

```
aws ec2 delete-transit-gateway-policy-table \  
  --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

Saída:

```
{  
  "TransitGatewayPolicyTables": [  
    {  
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",  
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",  
      "State": "deleting",  
      "CreationTime": "2023-11-28T16:36:43+00:00",  
      "Tags": []  
    }  
  ]  
}
```

Para obter mais informações, consulte as [tabelas de políticas do Transit Gateway](#) no Guia do usuário do Transit Gateway.

- Para API obter detalhes, consulte [DeleteTransitGatewayPolicyTable](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-prefix-list-reference

O código de exemplo a seguir mostra como usar `delete-transit-gateway-prefix-list-reference`.

AWS CLI

Para excluir uma referência de lista de prefixos

O `delete-transit-gateway-prefix-list-reference` exemplo a seguir exclui a referência da lista de prefixos especificada.

```
aws ec2 delete-transit-gateway-prefix-list-reference \  
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
  --prefix-list-id pl-1111112222222333
```

Saída:

```
{
  "TransitGatewayPrefixListReference": {
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-1111112222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "deleting",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}
```

Para obter mais informações, consulte as [referências da lista de prefixos](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DeleteTransitGatewayPrefixListReference](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-route-table

O código de exemplo a seguir mostra como usar delete-transit-gateway-route-table.

AWS CLI

Para excluir uma tabela de rotas do Transit Gateway

O delete-transit-gateway-route-table exemplo a seguir exclui a tabela de rotas do gateway de trânsito especificada.

```
aws ec2 delete-transit-gateway-route-table \
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE
```

Saída:

```
{
  "TransitGatewayRouteTable": {
    "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "State": "deleting",
```

```
    "DefaultAssociationRouteTable": false,  
    "DefaultPropagationRouteTable": false,  
    "CreationTime": "2019-07-17T20:27:26.000Z"  
  }  
}
```

Para obter mais informações, consulte [Excluir uma tabela de rotas do Transit Gateways](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DeleteTransitGatewayRouteTable](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-route

O código de exemplo a seguir mostra como usar `delete-transit-gateway-route`.

AWS CLI

Para excluir um CIDR bloco de uma tabela de rotas

O `delete-transit-gateway-route` exemplo a seguir exclui o CIDR bloco da tabela de rotas do gateway de trânsito especificada.

```
aws ec2 delete-transit-gateway-route \  
  --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \  
  --destination-cidr-block 10.0.2.0/24
```

Saída:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  
      {  
        "ResourceId": "vpc-0065acced4EXAMPLE",  
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
        "ResourceType": "vpc"  
      }  
    ],  
    "Type": "static",  
    "State": "deleted"  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Excluir uma rota estática](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DeleteTransitGatewayRoute](#) na Referência de AWS CLI Comandos.

delete-transit-gateway-vpc-attachment

O código de exemplo a seguir mostra como usar delete-transit-gateway-vpc-attachment.

AWS CLI

Para excluir um VPC anexo do Transit Gateway

O delete-transit-gateway-vpc-attachment exemplo a seguir exclui o VPC anexo especificado.

```
aws ec2 delete-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE
```

Saída:

```
{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "VpcId": "vpc-0065acced4f61c651",
    "VpcOwnerId": "111122223333",
    "State": "deleting",
    "CreationTime": "2019-07-17T16:04:27.000Z"
  }
}
```

Para obter mais informações, consulte [Excluir um VPC anexo](#) no Guia de Gateways de Trânsito.

- Para API obter detalhes, consulte [DeleteTransitGatewayVpcAttachment](#) na Referência de AWS CLI Comandos.

delete-transit-gateway

O código de exemplo a seguir mostra como usar delete-transit-gateway.

AWS CLI

Para excluir um gateway de trânsito

O `delete-transit-gateway` exemplo a seguir exclui o gateway de trânsito especificado.

```
aws ec2 delete-transit-gateway \  
--transit-gateway-id tgw-01f04542b2EXAMPLE
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Excluir um gateway de trânsito](#) no Guia de gateways de trânsito.

- Para API obter detalhes, consulte [DeleteTransitGateway](#) na Referência de AWS CLI Comandos.

delete-verified-access-endpoint

O código de exemplo a seguir mostra como usar `delete-verified-access-endpoint`.

AWS CLI

Para excluir um endpoint de acesso verificado

O `delete-verified-access-endpoint` exemplo a seguir exclui o endpoint de acesso verificado especificado.

```
aws ec2 delete-verified-access-endpoint \  
--verified-access-endpoint-id vae-066fac616d4d546f2
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Endpoints de acesso verificado](#) no Guia do usuário do AWS Verified Access.

- Para API obter detalhes, consulte [DeleteVerifiedAccessEndpoint](#) na Referência de AWS CLI Comandos.

delete-verified-access-group

O código de exemplo a seguir mostra como usar `delete-verified-access-group`.

AWS CLI

Para excluir um grupo de acesso verificado

O `delete-verified-access-group` exemplo a seguir exclui o grupo de acesso verificado especificado.

```
aws ec2 delete-verified-access-group \  
  --verified-access-group-id vagr-0dbe967baf14b7235
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Grupos de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DeleteVerifiedAccessGroup](#) na Referência de AWS CLI Comandos.

delete-verified-access-instance

O código de exemplo a seguir mostra como usar `delete-verified-access-instance`.

AWS CLI

Para excluir uma instância de acesso verificado

O `delete-verified-access-instance` exemplo a seguir exclui a instância de acesso verificado especificada.

```
aws ec2 delete-verified-access-instance \  
  --verified-access-instance-id vai-0ce000c0b7643abea
```

Saída:

```
{  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "Testing Verified Access",  
    "VerifiedAccessTrustProviders": [],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-26T01:00:18"  
  }  
}
```

Para obter mais informações, consulte [Instâncias de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DeleteVerifiedAccessInstance](#) na Referência de AWS CLI Comandos.

delete-verified-access-trust-provider

O código de exemplo a seguir mostra como usar `delete-verified-access-trust-provider`.

AWS CLI

Para excluir um provedor confiável de acesso verificado

O `delete-verified-access-trust-provider` exemplo a seguir exclui o provedor confiável de acesso verificado especificado.

```
aws ec2 delete-verified-access-trust-provider \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Saída:

```
{
```

```
"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"
}
}
```

Para obter mais informações, consulte [Provedores confiáveis para acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DeleteVerifiedAccessTrustProvider](#) na Referência de AWS CLI Comandos.

delete-volume

O código de exemplo a seguir mostra como usar `delete-volume`.

AWS CLI

Para excluir um volume

Este exemplo de comando exclui um volume disponível com o ID do volume `devol-049df61146c4d7901`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- Para API obter detalhes, consulte [DeleteVolume](#) na Referência de AWS CLI Comandos.

delete-vpc-endpoint-connection-notifications

O código de exemplo a seguir mostra como usar `delete-vpc-endpoint-connection-notifications`.

AWS CLI

Para excluir uma notificação de conexão de endpoint

Este exemplo exclui a notificação de conexão do endpoint especificada.

Comando:

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

Saída:

```
{
  "Unsuccessful": []
}
```

- Para API obter detalhes, consulte [DeleteVpcEndpointConnectionNotifications](#) na Referência de AWS CLI Comandos.

delete-vpc-endpoint-service-configurations

O código de exemplo a seguir mostra como usar `delete-vpc-endpoint-service-configurations`.

AWS CLI

Para excluir uma configuração de serviço de endpoint

Este exemplo exclui a configuração do serviço de endpoint especificada.

Comando:

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

Saída:

```
{
  "Unsuccessful": []
}
```

- Para API obter detalhes, consulte [DeleteVpcEndpointServiceConfigurations](#) na Referência de AWS CLI Comandos.

delete-vpc-endpoints

O código de exemplo a seguir mostra como usar `delete-vpc-endpoints`.

AWS CLI

Para excluir um endpoint

Este exemplo exclui os endpoints `vpce-aa22bb33` e `vpce-1a2b3c4d`. Se o comando for parcialmente bem-sucedido ou malsucedido, uma lista de itens malsucedidos será retornada. Se o comando for bem-sucedido, a lista retornada estará vazia.

Comando:

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

Saída:

```
{
  "Unsuccessful": []
}
```

- Para API obter detalhes, consulte [DeleteVpcEndpoints](#) na Referência de AWS CLI Comandos.

delete-vpc-peering-connection

O código de exemplo a seguir mostra como usar `delete-vpc-peering-connection`.

AWS CLI

Para excluir uma conexão de VPC emparelhamento

Este exemplo exclui a conexão de VPC emparelhamento especificada.

Comando:

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [DeleteVpcPeeringConnection](#) na Referência de AWS CLI Comandos.

delete-vpc

O código de exemplo a seguir mostra como usar `delete-vpc`.

AWS CLI

Para excluir um VPC

Este exemplo exclui o especificado VPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- Para API obter detalhes, consulte [DeleteVpc](#) na Referência de AWS CLI Comandos.

delete-vpn-connection-route

O código de exemplo a seguir mostra como usar `delete-vpn-connection-route`.

AWS CLI

Para excluir uma rota estática de uma VPN conexão

Este exemplo exclui a rota estática especificada da VPN conexão especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```


- Para API obter detalhes, consulte [DeleteVpnConnectionRoute](#) na Referência de AWS CLI Comandos.

delete-vpn-connection

O código de exemplo a seguir mostra como usar `delete-vpn-connection`.

AWS CLI

Para excluir uma VPN conexão

Este exemplo exclui a VPN conexão especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- Para API obter detalhes, consulte [DeleteVpnConnection](#) na Referência de AWS CLI Comandos.

delete-vpn-gateway

O código de exemplo a seguir mostra como usar `delete-vpn-gateway`.

AWS CLI

Para excluir um gateway privado virtual

Este exemplo exclui o gateway privado virtual especificado. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- Para API obter detalhes, consulte [DeleteVpnGateway](#) na Referência de AWS CLI Comandos.

deprovision-byoip-cidr

O código de exemplo a seguir mostra como usar `deprovision-byoip-cidr`.

AWS CLI

Para remover um intervalo de endereços IP do uso

O exemplo a seguir remove o intervalo de endereços especificado do uso com AWS.

```
aws ec2 deprovision-byoip-cidr \  
  --cidr 203.0.113.25/24
```

Saída:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "State": "pending-deprovision"  
  }  
}
```

- Para API obter detalhes, consulte [DeprovisionByoipCidr](#) na Referência de AWS CLI Comandos.

deprovision-ipam-pool-cidr

O código de exemplo a seguir mostra como usar `deprovision-ipam-pool-cidr`.

AWS CLI

Para desprovisionar um pool IPAM CIDR

O `deprovision-ipam-pool-cidr` exemplo a seguir desprovisiona um CIDR provisionado em um 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
```

Saída:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "11.0.0.0/16",  
    "State": "pending-deprovision"  
  }  
}
```

Para obter mais informações, consulte [Pool de desprovisionamento CIDRs](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [DeprovisionIpamPoolCidr](#) na Referência de AWS CLI Comandos.

deregister-image

O código de exemplo a seguir mostra como usar `deregister-image`.

AWS CLI

Para cancelar o registro de um AMI

Este exemplo cancela o registro especificado. AMI Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- Para API obter detalhes, consulte [DeregisterImage](#) na Referência de AWS CLI Comandos.

deregister-instance-event-notification-attributes

O código de exemplo a seguir mostra como usar `deregister-instance-event-notification-attributes`.

AWS CLI

Exemplo 1: Para remover todas as tags das notificações de eventos

O `deregister-instance-event-notification-attributes` exemplo a seguir remove `IncludeAllTagsOfInstance=true`, o que tem o efeito de `IncludeAllTagsOfInstance` definir como `false`.

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute IncludeAllTagsOfInstance=true
```

Saída:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [],  
    "IncludeAllTagsOfInstance": true  
  }  
}
```

Para obter mais informações, consulte [Eventos programados para suas instâncias](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: Para remover tags específicas das notificações de eventos

O `deregister-instance-event-notification-attributes` exemplo a seguir remove a tag especificada das tags incluídas nas notificações de eventos. Para descrever as tags restantes incluídas nas notificações de eventos, use `describe-instance-event-notification-attributes`.

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute InstanceTagKeys="tag-key2"
```

Saída:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [  
      "tag-key2"  
    ],  
    "IncludeAllTagsOfInstance": false  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Eventos programados para suas instâncias](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DeregisterInstanceEventNotificationAttributes](#) na Referência de AWS CLI Comandos.

deregister-transit-gateway-multicast-group-members

O código de exemplo a seguir mostra como usar `deregister-transit-gateway-multicast-group-members`.

AWS CLI

Para cancelar o registro de membros de um grupo multicast

Este exemplo cancela o registro do membro do grupo de interface de rede especificado do grupo multicast do gateway de trânsito.

```
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
```

Saída:

```
{
  "DeregisteredMulticastGroupMembers": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
    "RegisteredNetworkInterfaceIds": [
      "eni-0e246d3269EXAMPLE"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Para obter mais informações, consulte [Cancelar o registro de membros de um grupo multicast](#) no Guia do usuário do AWS Transit Gateways.

- Para API obter detalhes, consulte [DeregisterTransitGatewayMulticastGroupMembers](#) na Referência de AWS CLI Comandos.

deregister-transit-gateway-multicast-group-source

O código de exemplo a seguir mostra como usar `deregister-transit-gateway-multicast-group-source`.

AWS CLI

Para cancelar o registro de uma origem do grupo multicast do Transit Gateway

Este exemplo cancela o registro da origem do grupo de interface de rede especificado do grupo multicast.

```
aws ec2 deregister-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
```

Saída:

```
{
  "DeregisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "DeregisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Para obter mais informações, consulte [Cancelar o registro de fontes de um grupo multicast](#) no Guia do usuário do AWS Transit Gateways.

- Para API obter detalhes, consulte [DeregisterTransitGatewayMulticastGroupSource](#) na Referência de AWS CLI Comandos.

describe-account-attributes

O código de exemplo a seguir mostra como usar `describe-account-attributes`.

AWS CLI

Para descrever todos os atributos da sua AWS conta

Este exemplo descreve os atributos da sua AWS conta.

Comando:

```
aws ec2 describe-account-attributes
```

Saída:

```
{
  "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"
      }
    ]
  }
]
```

Para descrever um único atributo para sua AWS conta

Este exemplo descreve o `supported-platforms` atributo da sua AWS conta.

Comando:

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

Saída:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "supported-platforms",
      "AttributeValues": [
        {
          "AttributeValue": "EC2"
        },
        {
          "AttributeValue": "VPC"
        }
      ]
    }
  ]
}
```



```

    ]
  }
]
}

```

- Para API obter detalhes, consulte [DescribeAccountAttributes](#) na Referência de AWS CLI Comandos.

describe-address-transfers

O código de exemplo a seguir mostra como usar `describe-address-transfers`.

AWS CLI

Para descrever uma transferência de endereço IP elástico

O `describe-address-transfers` exemplo a seguir descreve a transferência do endereço IP elástico para o endereço IP elástico especificado.

```

aws ec2 describe-address-transfers \
  --allocation-ids eipalloc-09ad461b0d03f6aaf

```

Saída:

```

{
  "AddressTransfers": [
    {
      "PublicIp": "100.21.184.216",
      "AllocationId": "eipalloc-09ad461b0d03f6aaf",
      "TransferAccountId": "123456789012",
      "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",
      "AddressTransferStatus": "pending"
    }
  ]
}

```

Para obter mais informações, consulte [Transferir endereços IP elásticos](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeAddressTransfers](#) na Referência de AWS CLI Comandos.

describe-addresses-attribute

O código de exemplo a seguir mostra como usar `describe-addresses-attribute`.

AWS CLI

Para visualizar os atributos do nome de domínio associado a um endereço IP elástico

Os `describe-addresses-attribute` exemplos a seguir retornam os atributos do nome de domínio associado ao endereço IP elástico.

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
```

Saída:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.com."  
    }  
  ]  
}
```

Para visualizar os atributos de um endereço IP elástico, você deve primeiro associar um nome de domínio ao endereço IP elástico. Para obter mais informações, consulte [Use reverse DNS para aplicativos de e-mail](#) no Amazon EC2 User Guide ou [modify-address-attribute](#) na AWS CLI Command Reference.

- Para API obter detalhes, consulte [DescribeAddressesAttribute](#) na Referência de AWS CLI Comandos.

describe-addresses

O código de exemplo a seguir mostra como usar `describe-addresses`.

AWS CLI

Exemplo 1: recuperar detalhes de todos os seus endereços IP elásticos

O exemplo `describe-addresses` a seguir mostra os detalhes dos seus endereços IP elásticos.

```
aws ec2 describe-addresses
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: Para recuperar detalhes de seus endereços IP elásticos para EC2 - VPC

O `describe-addresses` exemplo a seguir exibe detalhes sobre seus endereços IP elásticos para uso com instâncias em um VPC.

```
aws ec2 describe-addresses \
```

```
--filters "Name=domain,Values=vpc"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 3: recuperar detalhes sobre um endereço IP elástico especificado pelo ID de alocação

O `describe-addresses` exemplo a seguir exibe detalhes sobre o endereço IP elástico com o ID de alocação especificado, que está associado a uma instância em EC2 -VPC.

```
aws ec2 describe-addresses \
  --allocation-ids eipalloc-282d9641
```

Saída:

```
{
  "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"
    }
  ]
}
```

```

    }
  ]
}

```

Exemplo 4: Para recuperar detalhes sobre um endereço IP elástico especificado por seu endereço IP VPC privado

O `describe-addresses` exemplo a seguir exibe detalhes sobre o endereço IP elástico associado a um determinado endereço IP privado em EC2 -VPC.

```

aws ec2 describe-addresses \
  --filters "Name=private-ip-address,Values=10.251.50.12"

```

Exemplo 5: Para recuperar detalhes sobre endereços IP elásticos em EC2 -Classic

O `describe-addresses` exemplo a seguir exibe detalhes sobre seus endereços IP elásticos para uso no EC2 -Classic.

```

aws ec2 describe-addresses \
  --filters "Name=domain,Values=standard"

```

Saída:

```

{
  "Addresses": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "PublicIp": "203.0.110.25",
      "PublicIpv4Pool": "amazon",
      "Domain": "standard"
    }
  ]
}

```

Exemplo 6: recuperar detalhes sobre um endereço IP elástico especificado pelo endereço IP público

O `describe-addresses` exemplo a seguir exibe detalhes sobre o endereço IP elástico com o valor `203.0.110.25`, que está associado a uma instância em EC2 -Classic.

```

aws ec2 describe-addresses \

```

```
--public-ips 203.0.110.25
```

Saída:

```
{
  "Addresses": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "PublicIp": "203.0.110.25",
      "PublicIpv4Pool": "amazon",
      "Domain": "standard"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeAddresses](#) na Referência de AWS CLI Comandos.

describe-aggregate-id-format

O código de exemplo a seguir mostra como usar `describe-aggregate-id-format`.

AWS CLI

Para descrever as configurações de formato de ID mais longas para todos os tipos de recursos em uma região

O `describe-aggregate-id-format` exemplo a seguir descreve o status geral do formato de ID longo para a região atual. O `Deadline` valor indica que os prazos para que esses recursos mudassem permanentemente do formato de ID curto para o formato de ID longo expiraram. O `UseLongIdsAggregated` valor indica que todos os IAM usuários e IAM funções estão configurados para usar o formato de ID longo para todos os tipos de recursos.

```
aws ec2 describe-aggregate-id-format
```

Saída:

```
{
  "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
  },
  ...
]
}

```

- Para API obter detalhes, consulte [DescribeAggregateldFormat](#) na Referência de AWS CLI Comandos.

describe-availability-zones

O código de exemplo a seguir mostra como usar `describe-availability-zones`.

AWS CLI

Para descrever suas zonas de disponibilidade

O exemplo `describe-availability-zones` a seguir exibe detalhes das zonas de disponibilidade disponíveis para você. A resposta inclui zonas de disponibilidade somente para a região atual. Neste exemplo, ela usa a região padrão dos perfis do `us-west-2` (Oregon).

```
aws ec2 describe-availability-zones
```

Saída:

```

{
  "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"
      }
    ]
  }
```

- Para API obter detalhes, consulte [DescribeAvailabilityZones](#) na Referência de AWS CLI Comandos.

describe-aws-network-performance-metric-subscription

O código de exemplo a seguir mostra como usar `describe-aws-network-performance-metric-subscription`.

AWS CLI

Para descrever suas assinaturas métricas

O `describe-aws-network-performance-metric-subscriptions` exemplo a seguir descreve suas assinaturas métricas.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Saída:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciar assinaturas no Guia](#) do usuário de desempenho da infraestrutura.

- Para API obter detalhes, consulte [DescribeAwsNetworkPerformanceMetricSubscription](#) na Referência de AWS CLI Comandos.

describe-aws-network-performance-metric-subscriptions

O código de exemplo a seguir mostra como usar `describe-aws-network-performance-metric-subscriptions`.

AWS CLI

Para descrever suas assinaturas métricas

O `describe-aws-network-performance-metric-subscriptions` exemplo a seguir descreve suas assinaturas métricas.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Saída:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciar assinaturas no Guia](#) do usuário de desempenho da infraestrutura.

- Para API obter detalhes, consulte [DescribeAwsNetworkPerformanceMetricSubscriptions](#) na Referência de AWS CLI Comandos.

describe-bundle-tasks

O código de exemplo a seguir mostra como usar `describe-bundle-tasks`.

AWS CLI

Para descrever suas tarefas do pacote

Este exemplo descreve todas as suas tarefas do pacote.

Comando:

```
aws ec2 describe-bundle-tasks
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeBundleTasks](#) na Referência de AWS CLI Comandos.

describe-byoip-cidrs

O código de exemplo a seguir mostra como usar describe-byoip-cidrs.

AWS CLI

Para descrever seus intervalos de endereços provisionados

O describe-byoip-cidrs exemplo a seguir exibe detalhes sobre os intervalos de IPv4 endereços públicos que você provisionou para uso. AWS

```
aws ec2 describe-byoip-cidrs
```

Saída:

```
{
  "ByoipCidrs": [
    {
      "Cidr": "203.0.113.25/24",
      "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
      "State": "provisioned"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeByoipCidrs](#) na Referência de AWS CLI Comandos.

describe-capacity-reservation-fleets

O código de exemplo a seguir mostra como usar `describe-capacity-reservation-fleets`.

AWS CLI

Para ver uma frota de reserva de capacidade

O `describe-capacity-reservation-fleets` exemplo a seguir lista as informações de configuração e capacidade da frota de reserva de capacidade especificada. Ele também lista detalhes sobre as reservas de capacidade individuais que estão dentro da frota. :

```
aws ec2 describe-capacity-reservation-fleets \
  --capacity-reservation-fleet-ids crf-abcdef01234567890
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações sobre frotas de reserva de capacidade, consulte [Frotas de reserva de capacidade](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeCapacityReservationFleets](#) na Referência de AWS CLI Comandos.

describe-capacity-reservations

O código de exemplo a seguir mostra como usar `describe-capacity-reservations`.

AWS CLI

Exemplo 1: Para descrever uma ou mais de suas reservas de capacidade

O `describe-capacity-reservations` exemplo a seguir exibe detalhes sobre todas as suas reservas de capacidade na AWS região atual.

```
aws ec2 describe-capacity-reservations
```

Saída:

```
{
```

```

"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"
  }
]
}

```

Exemplo 2: Para descrever uma ou mais de suas reservas de capacidade

O `describe-capacity-reservations` exemplo a seguir exibe detalhes sobre a reserva de capacidade especificada.

```

aws ec2 describe-capacity-reservations \
  --capacity-reservation-ids cr-1234abcd56EXAMPLE

```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Visualizar uma reserva de capacidade](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeCapacityReservations](#) na Referência de AWS CLI Comandos.

describe-carrier-gateways

O código de exemplo a seguir mostra como usar `describe-carrier-gateways`.

AWS CLI

Para descrever todos os gateways da operadora

O `describe-carrier-gateways` exemplo a seguir lista todos os gateways de sua operadora.

```
aws ec2 describe-carrier-gateways
```

Saída:

```
{
  "CarrierGateways": [
    {
      "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
      "VpcId": "vpc-0c529aEXAMPLE",
      "State": "available",
      "OwnerId": "123456789012",
      "Tags": [
        {
          "Key": "example",
          "Value": "tag"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Carrier gateways](https://docs.aws.amazon.com/vpc/latest/userguide/carrier_gateway.html)< https://docs.aws.amazon.com/vpc/latest/userguide/carrier_gateway.html> no Guia do usuário da Amazon Virtual Private Cloud.

- Para API obter detalhes, consulte [DescribeCarrierGateways](#) na Referência de AWS CLI Comandos.

describe-classic-link-instances

O código de exemplo a seguir mostra como usar `describe-classic-link-instances`.

AWS CLI

Para descrever instâncias EC2 clássicas vinculadas

Este exemplo lista todas as suas instâncias EC2 -Classic vinculadas.

Comando:

```
aws ec2 describe-classic-link-instances
```

Saída:

```
{
  "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"
    }
  ]
}
]
```

Este exemplo lista todas as suas instâncias EC2 -Classic vinculadas e filtra a resposta para incluir somente instâncias vinculadas à VPC vpc-88888888.

Comando:

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

Saída:

```
{
```

```

    "Instances": [
      {
        "InstanceId": "i-1234567890abcdef0",
        "VpcId": "vpc-88888888",
        "Groups": [
          {
            "GroupId": "sg-11122233"
          }
        ],
        "Tags": [
          {
            "Value": "ClassicInstance",
            "Key": "Name"
          }
        ]
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [DescribeClassicLinkInstances](#) na Referência de AWS CLI Comandos.

describe-client-vpn-authorization-rules

O código de exemplo a seguir mostra como usar `describe-client-vpn-authorization-rules`.

AWS CLI

Para descrever as regras de autorização para um VPN endpoint do cliente

O `describe-client-vpn-authorization-rules` exemplo a seguir exibe detalhes sobre as regras de autorização para o VPN endpoint do cliente especificado.

```

aws ec2 describe-client-vpn-authorization-rules \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Saída:

```

{
  "AuthorizationRules": [

```

```

    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "GroupId": "",
      "AccessAll": true,
      "DestinationCidr": "0.0.0.0/0",
      "Status": {
        "Code": "active"
      }
    }
  ]
}

```

Para obter mais informações, consulte [Regras de autorização](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [DescribeClientVpnAuthorizationRules](#) na Referência de AWS CLI Comandos.

describe-client-vpn-connections

O código de exemplo a seguir mostra como usar describe-client-vpn-connections.

AWS CLI

Para descrever as conexões com um VPN endpoint do cliente

O describe-client-vpn-connections exemplo a seguir exibe detalhes sobre as conexões do cliente com o VPN endpoint do cliente especificado.

```

aws ec2 describe-client-vpn-connections \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Saída:

```

{
  "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": "-"
  }
]
}

```

Para obter mais informações, consulte [Conexões do AWS cliente](#) no Guia do VPN administrador do cliente.

- Para API obter detalhes, consulte [DescribeClientVpnConnections](#) na Referência de AWS CLI Comandos.

describe-client-vpn-endpoints

O código de exemplo a seguir mostra como usar describe-client-vpn-endpoints.

AWS CLI

Para descrever os VPN endpoints do seu cliente

O `describe-client-vpn-endpoints` exemplo a seguir exibe detalhes sobre todos os VPN endpoints do seu cliente.

```
aws ec2 describe-client-vpn-endpoints
```

Saída:

```
{
  "ClientVpnEndpoints": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "Description": "Endpoint for Admin access",
      "Status": {
        "Code": "available"
      },
      "CreationTime": "2020-11-13T11:37:27",
      "DnsName": "*.cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com",
      "ClientCidrBlock": "172.31.0.0/16",
      "DnsServers": [
        "8.8.8.8"
      ],
      "SplitTunnel": false,
      "VpnProtocol": "openvpn",
      "TransportProtocol": "udp",
      "VpnPort": 443,
      "ServerCertificateArn": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "AuthenticationOptions": [
        {
          "Type": "certificate-authentication",
          "MutualAuthentication": {
            "ClientRootCertificateChain": "arn:aws:acm:ap-
south-1:123456789012:certificate/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
          }
        }
      ],
      "ConnectionLogOptions": {
        "Enabled": true,
        "CloudwatchLogGroup": "Client-vpn-connection-logs",
        "CloudwatchLogStream": "cvpn-endpoint-123456789123abcde-ap-
south-1-2020/11/13-FCD8HEMVAcCw"
      },
    }
  ]
}
```

```

    "Tags": [
      {
        "Key": "Name",
        "Value": "Client VPN"
      }
    ],
    "SecurityGroupIds": [
      "sg-aabbcc11223344567"
    ],
    "VpcId": "vpc-a87f92c1",
    "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/
endpoints/cvpn-endpoint-123456789123abcde",
    "ClientConnectOptions": {
      "Enabled": false
    }
  }
]
}

```

Para obter mais informações, consulte [Client VPN Endpoints](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [DescribeClientVpnEndpoints](#) na Referência de AWS CLI Comandos.

describe-client-vpn-routes

O código de exemplo a seguir mostra como usar `describe-client-vpn-routes`.

AWS CLI

Para descrever as rotas para um VPN endpoint do cliente

O `describe-client-vpn-routes` exemplo a seguir exibe detalhes sobre as rotas para o VPN endpoint do Cliente especificado.

```

aws ec2 describe-client-vpn-routes \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Saída:

```
{
```

```

"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"
    }
  }
]
}

```

Para obter mais informações, consulte [Rotas](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [DescribeClientVpnRoutes](#) na Referência de AWS CLI Comandos.

describe-client-vpn-target-networks

O código de exemplo a seguir mostra como usar `describe-client-vpn-target-networks`.

AWS CLI

Para descrever as redes de destino para um VPN endpoint do cliente

O `describe-client-vpn-target-networks` exemplo a seguir exibe detalhes sobre as redes de destino para o VPN endpoint do cliente especificado.

```

aws ec2 describe-client-vpn-target-networks \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Saída:

```
{
  "ClientVpnTargetNetworks": [
    {
      "AssociationId": "cvpn-assoc-012e837060753dc3d",
      "VpcId": "vpc-11111222222333333",
      "TargetNetworkId": "subnet-0123456789abcabca",
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "Status": {
        "Code": "associating"
      },
      "SecurityGroups": [
        "sg-012345678910abcab"
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Target Networks](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [DescribeClientVpnTargetNetworks](#) na Referência de AWS CLI Comandos.

describe-coip-pools

O código de exemplo a seguir mostra como usar `describe-coip-pools`.

AWS CLI

Para descrever os pools de endereços IP de propriedade do cliente

O `describe-coip-pools` exemplo a seguir descreve os pools de endereços IP de propriedade do cliente em sua AWS conta.

```
aws ec2 describe-coip-pools
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Endereços IP pertencentes ao cliente](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [DescribeCoipPools](#) na Referência de AWS CLI Comandos.

describe-conversion-tasks

O código de exemplo a seguir mostra como usar `describe-conversion-tasks`.

AWS CLI

Para visualizar o status de uma tarefa de conversão

Este exemplo retorna o status de uma tarefa de conversão com o ID `import-i-ffvko9js`.

Comando:

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

Saída:

```

{
  "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"
}
]
}

```

- Para API obter detalhes, consulte [DescribeConversionTasks](#) na Referência de AWS CLI Comandos.

describe-customer-gateways

O código de exemplo a seguir mostra como usar `describe-customer-gateways`.

AWS CLI

Para descrever os gateways de seus clientes

Este exemplo descreve os gateways do seu cliente.

Comando:

```
aws ec2 describe-customer-gateways
```

Saída:

```
{
```

```
"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"  
  }  
]  
}
```

Para descrever um gateway de cliente específico

Este exemplo descreve o gateway do cliente especificado.

Comando:

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

Saída:

```
{  
  "CustomerGateways": [  
    {  
      "CustomerGatewayId": "cgw-0e11f167",  
      "IpAddress": "12.1.2.3",  
      "State": "available",  
      "Type": "ipsec.1",  
      "BgpAsn": "65534"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeCustomerGateways](#) na Referência de AWS CLI Comandos.

describe-dhcp-options

O código de exemplo a seguir mostra como usar `describe-dhcp-options`.

AWS CLI

Exemplo 1: Para descrever suas DHCP opções

O `describe-dhcp-options` exemplo a seguir recupera detalhes sobre suas DHCP opções.

```
aws ec2 describe-dhcp-options
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Trabalhando com conjuntos de DHCP opções](#) no Guia AWS VPC do usuário.

Exemplo 2: Para descrever suas DHCP opções e filtrar a saída

O `describe-dhcp-options` exemplo a seguir descreve suas DHCP opções e usa um filtro para retornar somente DHCP as opções que você tem `example.com` para o servidor de nomes de domínio. O exemplo usa o `--query` parâmetro para exibir somente as informações de configuração e o ID na saída.

```

aws ec2 describe-dhcp-options \
  --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
  --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"

```

Saída:

```

[
  [
    [
      {
        "Key": "domain-name",
        "Values": [
          {

```

```

        "Value": "example.com"
      }
    ]
  },
  {
    "Key": "domain-name-servers",
    "Values": [
      {
        "Value": "172.16.16.16"
      }
    ]
  }
],
"dopt-001122334455667ab"
]

```

Para obter mais informações, consulte [Trabalhando com conjuntos de DHCP opções](#) no Guia AWS VPC do usuário.

- Para API obter detalhes, consulte [DescribeDhcpOptions](#) na Referência de AWS CLI Comandos.

describe-egress-only-internet-gateways

O código de exemplo a seguir mostra como usar `describe-egress-only-internet-gateways`.

AWS CLI

Para descrever seus gateways de Internet somente de saída

Este exemplo descreve seus gateways de Internet somente de saída.

Comando:

```
aws ec2 describe-egress-only-internet-gateways
```

Saída:

```

{
  "EgressOnlyInternetGateways": [
    {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [

```

```
    {
      "State": "attached",
      "VpcId": "vpc-0c62a468"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeEgressOnlyInternetGateways](#) na Referência de AWS CLI Comandos.

describe-elastic-gpus

O código de exemplo a seguir mostra como usar `describe-elastic-gpus`.

AWS CLI

Para descrever um elástico GPU

Comando:

```
aws ec2 describe-elastic-gpus --elastic-gpu-ids egpu-12345678901234567890abcdefghijkl
```

- Para API obter detalhes, consulte [DescribeElasticGpus](#) na Referência de AWS CLI Comandos.

describe-export-image-tasks

O código de exemplo a seguir mostra como usar `describe-export-image-tasks`.

AWS CLI

Para monitorar uma tarefa de exportação de imagem

O `describe-export-image-tasks` exemplo a seguir verifica o status da tarefa de exportação de imagem especificada. O arquivo de imagem resultante no Amazon S3 é `my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk`

```
aws ec2 describe-export-image-tasks \
  --export-image-task-ids export-ami-1234567890abcdef0
```

Saída para uma tarefa de exportação de imagem que está em andamento.

```
{
  "ExportImageTasks": [
    {
      "ExportImageTaskId": "export-ami-1234567890abcdef0"
      "Progress": "21",
      "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
      },
      "Status": "active",
      "StatusMessage": "updating"
    }
  ]
}
```

Saída para uma tarefa de exportação de imagem concluída.

```
{
  "ExportImageTasks": [
    {
      "ExportImageTaskId": "export-ami-1234567890abcdef0"
      "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
      },
      "Status": "completed"
    }
  ]
}
```

Para obter mais informações, consulte [Exportar uma VM de uma AMI no Guia do](#) usuário do VM Import/Export.

- Para API obter detalhes, consulte [DescribeExportImageTasks](#) na Referência de AWS CLI Comandos.

describe-export-tasks

O código de exemplo a seguir mostra como usar describe-export-tasks.

AWS CLI

Para listar detalhes sobre uma tarefa de exportação de instância

Este exemplo descreve a tarefa de exportação com ID `export-i-fh8sjjsq`.

Comando:

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeExportTasks](#) na Referência de AWS CLI Comandos.

`describe-fast-launch-images`

O código de exemplo a seguir mostra como usar `describe-fast-launch-images`.

AWS CLI

Para descrever os detalhes do Windows AMIs que estão configurados para um lançamento mais rápido

O `describe-fast-launch-images` exemplo a seguir descreve os detalhes de cada um dos itens AMIs em sua conta que estão configurados para lançamento mais rápido, incluindo o tipo de recurso, a configuração do instantâneo, os detalhes do modelo de execução, o número máximo de inicializações paralelas, o ID do AMI proprietário, o estado da configuração de inicialização rápida, o motivo pelo qual o estado foi alterado e a hora em que a mudança de estado ocorreu.

```
aws ec2 describe-fast-launch-images
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações sobre como configurar um Windows AMI para um lançamento mais rápido, consulte [Configurar seu AMI para um lançamento mais rápido](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DescribeFastLaunchImages](#) na Referência de AWS CLI Comandos.

describe-fast-snapshot-restores

O código de exemplo a seguir mostra como usar `describe-fast-snapshot-restores`.

AWS CLI

Para descrever restaurações rápidas de instantâneos

O `describe-fast-snapshot-restores` exemplo a seguir exibe detalhes de todas as restaurações rápidas de instantâneos com um estado de `disabled`

```
aws ec2 describe-fast-snapshot-restores \
  --filters Name=state,Values=disabled
```

Saída:

```
{
  "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"
    }
  ]
}
```

O `describe-fast-snapshot-restores` exemplo a seguir descreve todas as restaurações rápidas de instantâneos.

```
aws ec2 describe-fast-snapshot-restores
```

- Para API obter detalhes, consulte [DescribeFastSnapshotRestores](#) na Referência de AWS CLI Comandos.

describe-fleet-history

O código de exemplo a seguir mostra como usar `describe-fleet-history`.

AWS CLI

Para descrever o histórico da EC2 frota

O `describe-fleet-history` exemplo a seguir retorna o histórico da EC2 frota especificada começando no horário especificado. O resultado é para uma EC2 frota com duas instâncias em execução.

```
aws ec2 describe-fleet-history \  
  --fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --start-time 2020-09-01T00:00:00Z
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [Gerenciando uma EC2 frota](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeFleetHistory](#) na Referência de AWS CLI Comandos.

describe-fleet-instances

O código de exemplo a seguir mostra como usar `describe-fleet-instances`.

AWS CLI

Para descrever as instâncias em execução de uma EC2 frota

O `describe-fleet-instances` exemplo a seguir descreve as instâncias em execução para a EC2 frota especificada.

```
aws ec2 describe-fleet-instances \
  --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

Saída:

```
{
```

```

"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"
}

```

Para obter mais informações, consulte [Gerenciando uma EC2 frota](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeFleetInstances](#) na Referência de AWS CLI Comandos.

describe-fleets

O código de exemplo a seguir mostra como usar `describe-fleets`.

AWS CLI

Para descrever uma EC2 frota

O `describe-fleets` exemplo a seguir descreve a EC2 frota especificada.

```

aws ec2 describe-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE

```

Saída:

```

{
  "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"
}
}
]
}

```

Para obter mais informações, consulte [Gerenciando uma EC2 frota](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeFleets](#) na Referência de AWS CLI Comandos.

describe-flow-logs

O código de exemplo a seguir mostra como usar `describe-flow-logs`.

AWS CLI

Exemplo 1: Para descrever todos os seus registros de fluxo

O `describe-flow-logs` exemplo a seguir exibe detalhes de todos os seus registros de fluxo.

```
aws ec2 describe-flow-logs
```

Saída:

```
{
  "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}"
    }
  ]
}
```



```
    ]
  }
```

Exemplo 2: Para descrever um subconjunto dos seus registros de fluxo

O `describe-flow-logs` exemplo a seguir usa um filtro para exibir detalhes somente dos registros de fluxo que estão no grupo de registros especificado no Amazon CloudWatch Logs.

```
aws ec2 describe-flow-logs \
  --filter "Name=log-group-name,Values=MyFlowLogs"
```

- Para API obter detalhes, consulte [DescribeFlowLogs](#) na Referência de AWS CLI Comandos.

describe-fpga-image-attribute

O código de exemplo a seguir mostra como usar `describe-fpga-image-attribute`.

AWS CLI

Para descrever os atributos de uma FPGA imagem da Amazon

Este exemplo descreve as permissões de carregamento para o especificado AFI.

Comando:

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute LoadPermission
```

Saída:

```
{
  "FpgaImageAttribute": {
    "FpgaImageId": "afi-0d123e123bfc85abc",
    "LoadPermissions": [
      {
        "UserId": "123456789012"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [DescribeFpgaImageAttribute](#) na Referência de AWS CLI Comandos.

describe-fpga-images

O código de exemplo a seguir mostra como usar `describe-fpga-images`.

AWS CLI

Para descrever FPGA imagens da Amazon

Este exemplo descreve AFIs que são de propriedade da conta123456789012.

Comando:

```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [DescribeFpgaImages](#) na Referência de AWS CLI Comandos.

describe-host-reservation-offerings

O código de exemplo a seguir mostra como usar `describe-host-reservation-offerings`.

AWS CLI

Para descrever as ofertas de reserva de anfitriões dedicados

Este exemplo descreve as reservas de host dedicado para a família de instâncias M4 que estão disponíveis para compra.

Comando:

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

Saída:

```
{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [DescribeHostReservationOfferings](#) na Referência de AWS CLI Comandos.

describe-host-reservations

O código de exemplo a seguir mostra como usar `describe-host-reservations`.

AWS CLI

Para descrever as reservas de anfitriões dedicados em sua conta

Este exemplo descreve as reservas de anfitriões dedicados em sua conta.

Comando:

```
aws ec2 describe-host-reservations
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeHostReservations](#) na Referência de AWS CLI Comandos.

describe-hosts

O código de exemplo a seguir mostra como usar describe-hosts.

AWS CLI

Para ver detalhes sobre hosts dedicados

O describe-hosts exemplo a seguir exibe detalhes dos hosts available dedicados em sua AWS conta.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Visualização de hosts dedicados](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeHosts](#) na Referência de AWS CLI Comandos.

describe-iam-instance-profile-associations

O código de exemplo a seguir mostra como usar `describe-iam-instance-profile-associations`.

AWS CLI

Para descrever associações de perfis de IAM instância

Este exemplo descreve todas as associações do perfil da sua IAM instância.

Comando:

```
aws ec2 describe-iam-instance-profile-associations
```

Saída:

```
{
  "IamInstanceProfileAssociations": [
    {
      "InstanceId": "i-09eb09efa73ec1dee",
      "State": "associated",
      "AssociationId": "iip-assoc-0db249b1f25fa24b8",
      "IamInstanceProfile": {
        "Id": "AIPAJVQN4F5WVLGCJDRGM",
        "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
    },
    {
      "InstanceId": "i-0402909a2f4dffd14",
      "State": "associating",
      "AssociationId": "iip-assoc-0d1ec06278d29f44a",
      "IamInstanceProfile": {
        "Id": "AGJAJVQN4F5WVLGCJABCM",
        "Arn": "arn:aws:iam::123456789012:instance-profile/user1-role"
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeInstanceProfileAssociations](#) na Referência de AWS CLI Comandos.

describe-id-format

O código de exemplo a seguir mostra como usar `describe-id-format`.

AWS CLI

Exemplo 1: Para descrever o formato de ID de um recurso

O `describe-id-format` exemplo a seguir descreve o formato de ID para grupos de segurança.

```
aws ec2 describe-id-format \
  --resource security-group
```

No exemplo de saída a seguir, o `Deadline` valor indica que o prazo para esse tipo de recurso mudar permanentemente do formato de ID curto para o formato de ID longo expirou às 00:00 de 15 de UTC agosto de 2018.

```
{
  "Statuses": [
    {
      "Deadline": "2018-08-15T00:00:00.000Z",
      "Resource": "security-group",
      "UseLongIds": true
    }
  ]
}
```

Exemplo 2: Para descrever o formato de ID de todos os recursos

O `describe-id-format` exemplo a seguir descreve o formato de ID para todos os tipos de recursos. Todos os tipos de recursos compatíveis com o formato de ID curto foram alterados para o formato de ID longo.

```
aws ec2 describe-id-format
```

- Para API obter detalhes, consulte [DescribeIdFormat](#) na Referência de AWS CLI Comandos.

describe-identity-id-format

O código de exemplo a seguir mostra como usar `describe-identity-id-format`.

AWS CLI

Para descrever o formato de ID de uma IAM função

O `describe-identity-id-format` exemplo a seguir descreve o formato de ID recebido pelas instâncias criadas pela IAM função `EC2Role` em sua AWS conta.

```
aws ec2 describe-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:role/my-iam-role \  
  --resource instance
```

O resultado a seguir indica que as instâncias criadas por essa função recebem IDs em formato de ID longo.

```
{  
  "Statuses": [  
    {  
      "Deadline": "2016-12-15T00:00:00Z",  
      "Resource": "instance",  
      "UseLongIds": true  
    }  
  ]  
}
```

Para descrever o formato de ID de um IAM usuário

O `describe-identity-id-format` exemplo a seguir descreve o formato de ID recebido pelos instantâneos criados pelo IAM usuário `AdminUser` em sua AWS conta.

```
aws ec2 describe-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource snapshot
```

A saída indica que os instantâneos criados por esse usuário são recebidos IDs no formato de ID longo.

```
{
```

```
"Statuses": [  
  {  
    "Deadline": "2016-12-15T00:00:00Z",  
    "Resource": "snapshot",  
    "UseLongIds": true  
  }  
]  
}
```

- Para API obter detalhes, consulte [DescribeIdentityIdFormat](#) na Referência de AWS CLI Comandos.

describe-image-attribute

O código de exemplo a seguir mostra como usar `describe-image-attribute`.

AWS CLI

Para descrever as permissões de lançamento de um AMI

Este exemplo descreve as permissões de lançamento para o especificado AMI.

Comando:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --  
attribute LaunchPermission
```

Saída:

```
{  
  "LaunchPermissions": [  
    {  
      "UserId": "123456789012"  
    }  
  ],  
  "ImageId": "ami-5731123e",  
}
```

Para descrever os códigos de produto para um AMI

Este exemplo descreve os códigos de produto para o especificado AMI. Observe que isso não AMI tem códigos de produto.

Comando:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

Saída:

```
{
  "ProductCodes": [],
  "ImageId": "ami-5731123e",
}
```

- Para API obter detalhes, consulte [DescribeImageAttribute](#) na Referência de AWS CLI Comandos.

describe-images

O código de exemplo a seguir mostra como usar describe-images.

AWS CLI

Exemplo 1: Para descrever um AMI

O describe-images exemplo a seguir descreve o especificado AMI na região especificada.

```
aws ec2 describe-images \
  --region us-east-1 \
  --image-ids ami-1234567890EXAMPLE
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Amazon Machine Images \(AMI\)](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para descrever AMIs com base em filtros

O `describe-images` exemplo a seguir descreve o Windows AMIs fornecido pela Amazon que é apoiado pela AmazonEBS.

```

aws ec2 describe-images \
  --owners amazon \
  --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"

```

Para obter um exemplo da saída de `describe-images`, consulte o Exemplo 1.

Para obter exemplos adicionais de uso de filtros, consulte [Listar e filtrar seus recursos](#) no Guia do EC2 usuário da Amazon.

Exemplo 3: Para descrever AMIs com base em tags

O `describe-images` exemplo a seguir descreve todos os AMIs que têm a `tagType=Custom`. O exemplo usa o `--query` parâmetro para exibir somente AMI IDs o.

```
aws ec2 describe-images \  
  --filters "Name=tag:Type,Values=Custom" \  
  --query 'Images[*].[ImageId]' \  
  --output text
```

Saída:

```
ami-1234567890EXAMPLE  
ami-0abcdef1234567890
```

Para obter mais exemplos de uso de filtros de tags, consulte Como [trabalhar com tags](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeImages](#) na Referência de AWS CLI Comandos.

`describe-import-image-tasks`

O código de exemplo a seguir mostra como usar `describe-import-image-tasks`.

AWS CLI

Para monitorar uma tarefa de importação de imagem

O `describe-import-image-tasks` exemplo a seguir verifica o status da tarefa de importação de imagem especificada.

```
aws ec2 describe-import-image-tasks \  
  --import-task-ids import-ami-1234567890abcdef0
```

Saída para uma tarefa de importação de imagem que está em andamento.

```
{  
  "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"
        }
    ]
}

```

Saída para uma tarefa de importação de imagem concluída. O ID do resultado AMI é fornecido por `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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeImportImageTasks](#) na Referência de AWS CLI Comandos.

describe-import-snapshot-tasks

O código de exemplo a seguir mostra como usar `describe-import-snapshot-tasks`.

AWS CLI

Para monitorar uma tarefa de importação de instantâneo

O `describe-import-snapshot-tasks` exemplo a seguir verifica o status da tarefa de importação instantânea especificada.

```
aws ec2 describe-import-snapshot-tasks \
  --import-task-ids import-snap-1234567890abcdef0
```

Saída para uma tarefa de importação de instantâneo que está em andamento:

```
{
  "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"
        }
      }
    }
  ]
}
```

Saída para uma tarefa de importação de instantâneo concluída. O ID do instantâneo resultante é fornecido por `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"
      }
    }
  }
]
}

```

- Para API obter detalhes, consulte [DescribeImportSnapshotTasks](#) na Referência de AWS CLI Comandos.

describe-instance-attribute

O código de exemplo a seguir mostra como usar `describe-instance-attribute`.

AWS CLI

Para descrever o tipo de instância

Este exemplo descreve o tipo de instância da instância especificada.

Comando:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute instanceType
```

Saída:

```
{
  "InstanceId": "i-1234567890abcdef0"
```



```
    "InstanceType": {  
      "Value": "t1.micro"  
    }  
  }  
}
```

Para descrever o `disableApiTermination` atributo

Este exemplo descreve o `disableApiTermination` atributo da instância especificada.

Comando:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute disableApiTermination
```

Saída:

```
{  
  "InstanceId": "i-1234567890abcdef0"  
  "DisableApiTermination": {  
    "Value": "false"  
  }  
}
```

Para descrever o mapeamento de dispositivos de blocos para uma instância

Este exemplo descreve o `blockDeviceMapping` atributo da instância especificada.

Comando:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute blockDeviceMapping
```

Saída:

```
{  
  "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"
        }
    }
],
}

```

- Para API obter detalhes, consulte [DescribeInstanceAttribute](#) na Referência de AWS CLI Comandos.

describe-instance-connect-endpoints

O código de exemplo a seguir mostra como usar `describe-instance-connect-endpoints`.

AWS CLI

Para descrever um endpoint do EC2 Instance Connect

O `describe-instance-connect-endpoints` exemplo a seguir descreve o EC2 Instance Connect Endpoint especificado.

```

aws ec2 describe-instance-connect-endpoints \
  --region us-east-1 \
  --instance-connect-endpoint-ids eice-0123456789example

```

Saída:

```

{
  "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": []
  }
]
}
```

Para obter mais informações, consulte [Create an EC2 Instance Connect Endpoint](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DescribeInstanceConnectEndpoints](#) na Referência de AWS CLI Comandos.

describe-instance-credit-specifications

O código de exemplo a seguir mostra como usar `describe-instance-credit-specifications`.

AWS CLI

Para descrever a opção de crédito para CPU uso de uma ou mais instâncias

O `describe-instance-credit-specifications` exemplo a seguir descreve a opção CPU de crédito para a instância especificada.

```
aws ec2 describe-instance-credit-specifications \
  --instance-ids i-1234567890abcdef0
```

Saída:

```
{
  "InstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CpuCredits": "unlimited"
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhar com instâncias de desempenho intermitentes](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DescribeInstanceCreditSpecifications](#) na Referência de AWS CLI Comandos.

describe-instance-event-notification-attributes

O código de exemplo a seguir mostra como usar `describe-instance-event-notification-attributes`.

AWS CLI

Para descrever as tags para notificações de eventos agendados

O `describe-instance-event-notification-attributes` exemplo a seguir descreve as tags que aparecerão nas notificações de eventos programados.

```
aws ec2 describe-instance-event-notification-attributes
```

Saída:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [],
    "IncludeAllTagsOfInstance": true
  }
}
```

Para obter mais informações, consulte [Eventos programados para suas instâncias](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeInstanceEventNotificationAttributes](#) na Referência de AWS CLI Comandos.

describe-instance-event-windows

O código de exemplo a seguir mostra como usar `describe-instance-event-windows`.

AWS CLI

Exemplo 1: Para descrever todas as janelas de eventos

O `describe-instance-event-windows` exemplo a seguir descreve todas as janelas de eventos na região especificada.

```
aws ec2 describe-instance-event-windows \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
}
```

Exemplo 2: Para descrever uma janela de evento específica

O `describe-instance-event-windows` exemplo a seguir descreve um evento específico usando o `instance-event-window` parâmetro para descrever uma janela de evento específica.

```
aws ec2 describe-instance-event-windows \  
  --region us-east-1 \  
  --instance-event-window-ids iew-0abcdef1234567890
```

Saída:

```
{  
  "InstanceEventWindows": [  
    {  
      "InstanceEventWindowId": "iew-0abcdef1234567890",  
      "Name": "myEventWindowName",  
      "CronExpression": "* 21-23 * * 2,3",  
      "AssociationTarget": {  
        "InstanceIds": [  
          "i-1234567890abcdef0",  
          "i-0598c7d356eba48d7"  
        ],  
        "Tags": [],  
        "DedicatedHostIds": []  
      },  
      "State": "active",  
      "Tags": []  
    }  
  ]  
}
```

Exemplo 3: Para descrever janelas de eventos que correspondem a um ou mais filtros

O `describe-instance-event-windows` exemplo a seguir descreve janelas de eventos que correspondem a um ou mais filtros usando o `filter` parâmetro. O `instance-id` filtro é usado para descrever todas as janelas de eventos associadas à instância especificada. Quando um filtro é usado, ele executa uma correspondência direta. No entanto, o `instance-id` é diferente. Se não houver correspondência direta com a ID da instância, ela retornará às associações indiretas com a janela do evento, como as tags da instância ou a ID do host dedicado (se a instância for um host dedicado).

```
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>
```

Saída:

```
{  
  "InstanceEventWindows": [  
    {  
      "InstanceEventWindowId": "iew-0dbc0adb66f235982",  
      "TimeRanges": [  
        {  
          "StartWeekDay": "sunday",  
          "StartHour": 2,  
          "EndWeekDay": "sunday",  
          "EndHour": 8  
        }  
      ],  
      "Name": "myEventWindowName",  
      "AssociationTarget": {  
        "InstanceIds": [],  
        "Tags": [],  
        "DedicatedHostIds": [  
          "h-0140d9a7ecbd102dd"  
        ]  
      },  
      "State": "active",  
      "Tags": []  
    }  
  ]  
}
```

No exemplo de saída, a instância está em um host dedicado, associado à janela de eventos.

Para restrições da janela de eventos, consulte [Considerações](#) no Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [DescribeInstanceEventWindows](#) na Referência de AWS CLI Comandos.

describe-instance-status

O código de exemplo a seguir mostra como usar `describe-instance-status`.

AWS CLI

Para descrever o status de uma instância

O exemplo `describe-instance-status` a seguir descreve o status atual da instância especificada.

```
aws ec2 describe-instance-status \  
  --instance-ids i-1234567890abcdef0
```

Saída:

```
{  
  "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"  
          }  
        ]  
      }  
    }  
  ]  
}
```



```

    }
  ]
}

```

Para obter mais informações, consulte [Monitore o status de suas instâncias](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeInstanceStatus](#) na Referência de AWS CLI Comandos.

describe-instance-topology

O código de exemplo a seguir mostra como usar `describe-instance-topology`.

AWS CLI

Para descrever a topologia de todas as suas instâncias

O `describe-instance-topology` exemplo a seguir descreve a topologia de todas as suas instâncias que correspondem aos tipos de instância compatíveis com esse comando.

```
aws ec2 describe-instance-topology \
  --region us-west-2
```

Saída:

```
{
  "Instances": [
    {
      "InstanceId": "i-111111111example",
      "InstanceType": "p4d.24xlarge",
      "GroupName": "my-ml-cpg",
      "NetworkNodes": [
        "nn-111111111example",
        "nn-222222222example",
        "nn-333333333example"
      ],
      "ZoneId": "usw2-az2",
      "AvailabilityZone": "us-west-2a"
    },
    {
      "InstanceId": "i-222222222example",
      "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"
}

```

Para obter mais informações, incluindo mais exemplos, consulte a [topologia de EC2 instâncias da Amazon](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DescribeInstanceTopology](#) na Referência de AWS CLI Comandos.

describe-instance-type-offerings

O código de exemplo a seguir mostra como usar describe-instance-type-offerings.

AWS CLI

Exemplo 1: Para listar os tipos de instância oferecidos em uma região

O `describe-instance-type-offerings` exemplo a seguir lista os tipos de instância oferecidos na região configurada como a região padrão para AWS CLI o.

```
aws ec2 describe-instance-type-offerings
```

Para listar os tipos de instância oferecidos em uma região diferente, especifique a região usando o `--region` parâmetro.

```
aws ec2 describe-instance-type-offerings \  
  --region us-east-2
```

Saída:

```
{  
  "InstanceTypeOfferings": [  
    {  
      "InstanceType": "m5.2xlarge",  
      "LocationType": "region",  
      "Location": "us-east-2"  
    },  
    {  
      "InstanceType": "t3.micro",  
      "LocationType": "region",  
      "Location": "us-east-2"  
    },  
    ...  
  ]  
}
```

Exemplo 2: Para listar os tipos de instância oferecidos em uma zona de disponibilidade

O `describe-instance-type-offerings` exemplo a seguir lista os tipos de instância oferecidos na zona de disponibilidade especificada. A zona de disponibilidade deve estar na região especificada.

```
aws ec2 describe-instance-type-offerings \  
  --location-type availability-zone \  
  --filters Name=location,Values=us-east-2a \  
  --region us-east-2
```

```
--region us-east-2
```

Exemplo 3: para verificar se um tipo de instância é compatível

O `describe-instance-type-offerings` comando a seguir indica se o tipo de `c5.xlarge` instância é compatível com a região especificada.

```
aws ec2 describe-instance-type-offerings \  
  --filters Name=instance-type,Values=c5.xlarge \  
  --region us-east-2
```

O `describe-instance-type-offerings` exemplo a seguir lista todos os tipos de instância C5 que são compatíveis com a região especificada.

```
aws ec2 describe-instance-type-offerings \  
  --filters Name=instance-type,Values=c5* \  
  --query "InstanceTypeOfferings[].InstanceType" \  
  --region us-east-2
```

Saída:

```
[  
  "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"  
]
```

- Para API obter detalhes, consulte [DescribeInstanceTypeOfferings](#) na Referência de AWS CLI Comandos.

describe-instance-types

O código de exemplo a seguir mostra como usar `describe-instance-types`.

AWS CLI

Exemplo 1: descrever um tipo de instância

O exemplo `describe-instance-types` a seguir exibe os detalhes do tipo da instância especificado.

```
aws ec2 describe-instance-types \  
--instance-types t2.micro
```

Saída:

```
{  
  "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
}
```

```
]
}
```

Para obter mais informações, consulte [Tipos de instância](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: filtrar os tipos de instância disponíveis

Você pode especificar um filtro para definir o escopo dos resultados para os tipos de instância que tenham uma característica específica. O exemplo `describe-instance-types` a seguir lista os tipos de instância compatíveis com hibernação.

```
aws ec2 describe-instance-types \
  --filters Name=hibernation-supported,Values=true --query
  'InstanceTypes[*].InstanceType'
```

Saída:

```
[
  "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"
]
```

Para obter mais informações, consulte [Tipos de instância](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [DescribeInstanceTypes](#) na Referência de AWS CLI Comandos.

describe-instances

O código de exemplo a seguir mostra como usar `describe-instances`.

AWS CLI

Exemplo 1: descrever uma instância

O exemplo `describe-instances` a seguir descreve a instância especificada.

```
aws ec2 describe-instances \  
  --instance-ids i-1234567890abcdef0
```

Saída:

```
{  
  "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-04a636d18e83cfac",
    "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": "11111111111111111111111111111111"
    },
    "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"
}
]
```

Exemplo 2: para filtrar instâncias com o tipo especificado

O exemplo `describe-instances` a seguir usa filtros para definir o escopo dos resultados para instâncias do tipo especificado.

```
aws ec2 describe-instances \  
  --filters Name=instance-type,Values=m5.Large
```

Para obter um exemplo da saída, consulte o Exemplo 1.

Para obter mais informações, consulte [Listar e filtrar usando o CLI](#) no Guia EC2 do usuário da Amazon.

Exemplo 3: filtrar instâncias com o tipo e a zona de disponibilidade especificados

O exemplo `describe-instances` a seguir usa vários filtros para definir o escopo dos resultados para instâncias com o tipo especificado que também estão na zona de disponibilidade especificada.

```
aws ec2 describe-instances \  
  --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-  
zone,Values=us-east-2c
```

Para obter um exemplo da saída, consulte o Exemplo 1.

Exemplo 4: Para filtrar instâncias com o tipo e a zona de disponibilidade especificados usando um JSON arquivo

O `describe-instances` exemplo a seguir usa um arquivo JSON de entrada para realizar a mesma filtragem do exemplo anterior. Quando os filtros ficam mais complicados, é mais fácil especificá-los em um JSON arquivo.

```
aws ec2 describe-instances \  
  --filters file://filters.json
```

Conteúdo de `filters.json`:

```
[  
  {  
    "Name": "instance-type",  
    "Values": ["t2.micro", "t3.micro"]
```

```
  },  
  {  
    "Name": "availability-zone",  
    "Values": ["us-east-2c"]  
  }  
]
```

Para obter um exemplo da saída, consulte o Exemplo 1.

Exemplo 5: filtrar instâncias com a tag Proprietário especificada

O exemplo `describe-instances` a seguir usa filtros de tag para definir o escopo dos resultados para instâncias que têm uma tag com a chave de tag especificada (Proprietário), independentemente do valor da tag.

```
aws ec2 describe-instances \  
  --filters "Name=tag-key,Values=Owner"
```

Para obter um exemplo da saída, consulte o Exemplo 1.

Exemplo 6: filtrar instâncias com o valor especificado my-team da tag

O exemplo `describe-instances` a seguir usa filtros de tag para definir o escopo dos resultados para instâncias que têm uma tag com o valor especificado da tag (my-team), independentemente da chave da tag.

```
aws ec2 describe-instances \  
  --filters "Name=tag-value,Values=my-team"
```

Para obter um exemplo da saída, consulte o Exemplo 1.

Exemplo 7: filtrar instâncias com a tag Proprietário e o valor my-team especificados

O exemplo `describe-instances` a seguir usa filtros de tag para definir o escopo dos resultados para instâncias que têm a tag especificada (Owner=my-team).

```
aws ec2 describe-instances \  
  --filters "Name=tag:Owner,Values=my-team"
```

Para obter um exemplo da saída, consulte o Exemplo 1.

Exemplo 8: Para exibir somente a instância e a sub-rede IDs para todas as instâncias

Os `describe-instances` exemplos a seguir usam o `--query` parâmetro para exibir somente a instância e a sub-rede de todas IDs as instâncias, em JSON formato.

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
```

Saída:

```
[  
  {  
    "Instance": "i-057750d42936e468a",  
    "Subnet": "subnet-069beee9b12030077"  
  },  
  {  
    "Instance": "i-001efd250faaa6ffa",  
    "Subnet": "subnet-0b715c6b7db68927a"  
  },  
  {  
    "Instance": "i-027552a73f021f3bd",  
    "Subnet": "subnet-0250c25a1f4e15235"  
  }  
  ...  
]
```

Exemplo 9: Para filtrar instâncias do tipo especificado e exibir somente sua instância IDs

O `describe-instances` exemplo a seguir usa filtros para definir o escopo dos resultados para instâncias do tipo especificado e o `--query` parâmetro para exibir somente a instância IDs.

```
aws ec2 describe-instances \  
  --filters "Name=instance-type,Values=t2.micro" \  
  --query "Reservations[*].Instances[*].[InstanceId]" \  
  --output json
```

```
--output text
```

Saída:

```
i-031c0dc19de2fb70c
i-00d8bfff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

Exemplo 10: filtrar instâncias do tipo especificado e exibir somente a instânciaIDs, a zona de disponibilidade e o valor da tag especificada

Os exemplos `describe-instances` a seguir exibem o ID da instância, a zona de disponibilidade e o valor da tag `Name` para instâncias que têm uma tag com o nome `tag-key`, em formato de tabela.

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
```

Saída:

```
-----
|                               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

Exemplo 11: descrever instâncias em um grupo com posicionamento em partições

O exemplo `describe-instances` a seguir descreve a instância especificada. A saída inclui as informações de posicionamento da instância, o que contém o nome do grupo de posicionamento e o número da partição da instância.

```
aws ec2 describe-instances \
  --instance-ids i-0123a456700123456 \
  --query "Reservations[*].Instances[*].Placement"
```

Saída:

```
[
  [
    {
      "AvailabilityZone": "us-east-1c",
      "GroupName": "HDFS-Group-A",
      "PartitionNumber": 3,
      "Tenancy": "default"
    }
  ]
]
```

Para obter mais informações, consulte [Descrição de instâncias em um grupo de posicionamento](#) no Guia EC2 do usuário da Amazon.

Exemplo 12: filtrar instâncias com o grupo de posicionamento e o número de partição especificados

O exemplo `describe-instances` a seguir filtra os resultados somente para as instâncias com o grupo de posicionamento e o número de partição especificados.

```
aws ec2 describe-instances \
```



```
--filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-partition-number,Values=7"
```

A seguir, são mostradas somente as informações relevantes da saída.

```
"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"  
    }  
  },  
]
```

Para obter mais informações, consulte [Descrição de instâncias em um grupo de posicionamento](#) no Guia EC2 do usuário da Amazon.

Exemplo 13: filtrar instâncias configuradas para permitir o acesso às tags dos metadados da instância

O exemplo `describe-instances` a seguir filtra os resultados somente para as instâncias que estão configuradas para permitir o acesso às tags de instância nos metadados da instância.

```
aws ec2 describe-instances \  
--filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \  
--query "Reservations[*].Instances[*].InstanceId" \  
--output text
```

A saída esperada é mostrada a seguir.

```
i-1234567890abcdefg
i-abcdefg1234567890
i-111111111aaaaaaaaa
i-aaaaaaaaa111111111
```

Para obter mais informações, consulte [Trabalhar com tags de instância em metadados de instância](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DescribeInstances](#) na Referência de AWS CLI Comandos.

describe-internet-gateways

O código de exemplo a seguir mostra como usar `describe-internet-gateways`.

AWS CLI

Para descrever um gateway de internet

O `describe-internet-gateways` exemplo a seguir descreve o gateway de internet especificado.

```
aws ec2 describe-internet-gateways \
  --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

Saída:

```
{
  "InternetGateways": [
    {
      "Attachments": [
        {
          "State": "available",
          "VpcId": "vpc-0a60eb65b4EXAMPLE"
        }
      ],
      "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
      "OwnerId": "123456789012",
      "Tags": [
        {
          "Key": "Name",
          "Value": "my-igw"
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

Para obter mais informações, consulte [Gateways de Internet](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DescribeInternetGateways](#) na Referência de AWS CLI Comandos.

describe-ipam-pools

O código de exemplo a seguir mostra como usar `describe-ipam-pools`.

AWS CLI

Para ver os detalhes de uma IPAM piscina

O `describe-ipam-pools` exemplo a seguir mostra os detalhes dos pools.

(Linux):

```
aws ec2 describe-ipam-pools \
  --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-
scope-02fc38cd4c48e7d38
```

(Windows):

```
aws ec2 describe-ipam-pools ^
  --filters Name=owner-id,Values=123456789012 Name=ipam-scope-id,Values=ipam-
scope-02fc38cd4c48e7d38
```

Saída:

```
{
  "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"
      }
    ]
  }
]
```

- Para API obter detalhes, consulte [DescribeIpamPools](#) na Referência de AWS CLI Comandos.

describe-ipam-resource-discoveries

O código de exemplo a seguir mostra como usar `describe-ipam-resource-discoveries`.

AWS CLI

Exemplo 1: Exibir detalhes completos das descobertas de recursos

Neste exemplo, você é um IPAM administrador delegado que deseja criar e compartilhar uma descoberta de recursos com o IPAM administrador em outra AWS organização para que o administrador possa gerenciar e monitorar os endereços IP dos recursos em sua organização.

Esse exemplo pode ser útil se:

Você tentou criar uma descoberta de recursos, mas recebeu um erro informando que atingiu o limite de 1. Você percebe que talvez já tenha criado uma descoberta de recursos e deseja visualizá-la em sua conta. Você tem recursos em uma região que não estão sendo descobertos pelo IPAM. Você quer ver o `--operating-regions` definido para o recurso e garantir que você tenha adicionado a região certa como uma região operacional para que os recursos lá possam ser descobertos.

O `describe-ipam-resource-discoveries` exemplo a seguir lista os detalhes da descoberta de recursos em sua AWS conta. Você pode ter uma descoberta de recursos por AWS região.

```
aws ec2 describe-ipam-resource-discoveries \
  --region us-east-1
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

Exemplo 2: Exibir somente a descoberta de recursos IDs

O `describe-ipam-resource-discoveries` exemplo a seguir lista o ID da descoberta do recurso em sua AWS conta. Você pode ter uma descoberta de recursos por AWS região.

```
aws ec2 describe-ipam-resource-discoveries \  
  --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \  
  --output text
```

Saída:

```
ipam-res-disco-0481e39b242860333
```

Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeIpamResourceDiscoveries](#) na Referência de AWS CLI Comandos.

describe-ipam-resource-discovery-associations

O código de exemplo a seguir mostra como usar `describe-ipam-resource-discovery-associations`.

AWS CLI

Para ver todas as associações de descoberta de recursos com seu IPAM

Neste exemplo, você é um administrador IPAM delegado que associou descobertas de recursos às suas IPAM para integrar outras contas às suas. IPAM Você percebeu que não IPAM está descobrindo os recursos nas regiões operacionais da descoberta de recursos conforme o esperado. Você deseja verificar o status e o estado da descoberta de recursos para garantir que a conta que a criou ainda esteja ativa e que a descoberta de recursos ainda esteja sendo compartilhada.

`--region` Deve ser a região de origem do seu IPAM.

O `describe-ipam-resource-discovery-associations` exemplo a seguir lista as associações de descoberta de recursos em sua AWS conta.

```
aws ec2 describe-ipam-resource-discovery-associations \  
  --region us-east-1
```

Saída:

```

{
  "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": []
    }
  ]
}

```

Neste exemplo, depois de executar esse comando, você percebe que tem uma descoberta de recurso não padrão ("IsDefault": false ``) that is ``"ResourceDiscoveryStatus": "not-found"e. "State": "create-complete" A conta do proprietário da descoberta de recursos foi encerrada. Se, em outro caso, você perceber que é "ResourceDiscoveryStatus": "not-found" e"State": "associate-complete", isso indica que uma das seguintes coisas aconteceu:

A descoberta do recurso foi excluída pelo proprietário da descoberta do recurso. O proprietário da descoberta do recurso cancelou o compartilhamento da descoberta do recurso.

Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeIpamResourceDiscoveryAssociations](#) na Referência de AWS CLI Comandos.

describe-ipam-scopes

O código de exemplo a seguir mostra como usar `describe-ipam-scopes`.

AWS CLI

Para ver os detalhes de um IPAM escopo

O `describe-ipam-scopes` exemplo a seguir mostra os detalhes dos escopos.

```
aws ec2 describe-ipam-scopes \
  --filters Name=owner-id,Values=123456789012 Name=ipam-
  id,Values=ipam-08440e7a3acde3908
```

Saída:

```
{
  "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"
      }
    ]
  }
]
}

```

- Para API obter detalhes, consulte [Descrever IPAM Scopes](#) na Referência de AWS CLI Comandos.

describe-ipams

O código de exemplo a seguir mostra como usar `describe-ipams`.

AWS CLI

Para ver os detalhes de um IPAM

O `describe-ipams` exemplo a seguir mostra os detalhes de um IPAM.

```
aws ec2 describe-ipams \
  --filters Name=owner-id,Values=123456789012
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [Describelpams](#) na Referência de AWS CLI Comandos.

describe-ipv6-pools

O código de exemplo a seguir mostra como usar `describe-ipv6-pools`.

AWS CLI

Para descrever seus grupos IPv6 de endereços

O `describe-ipv6-pools` exemplo a seguir exibe detalhes de todos os seus grupos de IPv6 endereços.

```
aws ec2 describe-ipv6-pools
```

Saída:

```
{
  "Ipv6Pools": [
    {
      "PoolId": "ipv6pool-ec2-012345abc12345abc",
      "PoolCidrBlocks": [
        {
          "Cidr": "2001:db8:123::/48"
        }
      ],
      "Tags": [
        {
          "Key": "pool-1",
          "Value": "public"
        }
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeIpv6Pools](#) na Referência de AWS CLI Comandos.

describe-key-pairs

O código de exemplo a seguir mostra como usar `describe-key-pairs`.

AWS CLI

Exibir um par de chaves

O exemplo `describe-key-pairs` a seguir mostra as informações do par de chaves especificado.

```
aws ec2 describe-key-pairs \  
  --key-names my-key-pair
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Descrever as chaves públicas](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeKeyPairs](#) na Referência de AWS CLI Comandos.

describe-launch-template-versions

O código de exemplo a seguir mostra como usar `describe-launch-template-versions`.

AWS CLI

Para descrever as versões do modelo de lançamento

Este exemplo descreve as versões do modelo de lançamento especificado.

Comando:

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

Saída:

```
{  
  "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"
}
```

- Para API obter detalhes, consulte [DescribeLaunchTemplateVersions](#) na Referência de AWS CLI Comandos.

describe-launch-templates

O código de exemplo a seguir mostra como usar `describe-launch-templates`.

AWS CLI

Para descrever os modelos de lançamento

Este exemplo descreve seus modelos de lançamento.

Comando:

aws ec2 describe-launch-templates

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLaunchTemplates](#) na Referência de AWS CLI Comandos.

describe-local-gateway-route-table-virtual-interface-group-associations

O código de exemplo a seguir mostra como usar `describe-local-gateway-route-table-virtual-interface-group-associations`.

AWS CLI

Para descrever associações entre grupos de interface virtual e tabelas de rotas de gateway local

O `describe-local-gateway-route-table-virtual-interface-group-associations` exemplo a seguir descreve as associações entre grupos de interface virtual e tabelas de rotas de gateway local em sua AWS conta.

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhando com gateways locais no Guia](#) do Usuário do AWS Outposts.

- Para API obter detalhes, consulte [DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations](#) na Referência de AWS CLI Comandos.

describe-local-gateway-route-table-vpc-associations

O código de exemplo a seguir mostra como usar `describe-local-gateway-route-table-vpc-associations`.

AWS CLI

Para descrever as associações entre VPCs as tabelas de rotas do gateway local

O `describe-local-gateway-route-table-vpc-associations` exemplo a seguir exibe informações sobre a associação especificada entre VPCs as tabelas de rotas do gateway local.

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
  --local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

Saída:

```
{
  "LocalGatewayRouteTableVpcAssociation": {
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af15EXAMPLE",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
    "VpcId": "vpc-0efe9bde08EXAMPLE",
    "State": "associated"
  }
}
```

[Para obter mais informações, consulte Tabelas de rotas do gateway local](#) no Guia do usuário do Outposts.

- Para API obter detalhes, consulte [DescribeLocalGatewayRouteTableVpcAssociations](#) na Referência de AWS CLI Comandos.

describe-local-gateway-route-tables

O código de exemplo a seguir mostra como usar `describe-local-gateway-route-tables`.

AWS CLI

Para descrever suas tabelas de rotas de gateway local

O `describe-local-gateway-route-tables` exemplo a seguir exibe detalhes sobre as tabelas de rotas do gateway local.

```
aws ec2 describe-local-gateway-route-tables
```

Saída:

```
{
  "LocalGatewayRouteTables": [
    {
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
      "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
      "State": "available"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLocalGatewayRouteTables](#) na Referência de AWS CLI Comandos.

`describe-local-gateway-virtual-interface-groups`

O código de exemplo a seguir mostra como usar `describe-local-gateway-virtual-interface-groups`.

AWS CLI

Para descrever grupos de interface virtual de gateway local

O `describe-local-gateway-virtual-interface-groups` exemplo a seguir descreve os grupos de interface virtual do gateway local em sua AWS conta.

```
aws ec2 describe-local-gateway-virtual-interface-groups
```

Saída:

```
{
  "LocalGatewayVirtualInterfaceGroups": [
    {
      "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
      "LocalGatewayVirtualInterfaceIds": [
        "lgw-vif-01a23bc4d5EXAMPLE",
        "lgw-vif-543ab21012EXAMPLE"
      ],
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "OwnerId": "123456789012",
      "Tags": []
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhando com gateways locais no Guia](#) do Usuário do AWS Outposts.

- Para API obter detalhes, consulte [DescribeLocalGatewayVirtualInterfaceGroups](#) na Referência de AWS CLI Comandos.

describe-local-gateway-virtual-interfaces

O código de exemplo a seguir mostra como usar `describe-local-gateway-virtual-interfaces`.

AWS CLI

Para descrever interfaces virtuais de gateway local

O `describe-local-gateway-virtual-interfaces` exemplo a seguir descreve as interfaces virtuais do gateway local em sua AWS conta.

```
aws ec2 describe-local-gateway-virtual-interfaces
```

Saída:

```
{
  "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": []
    }
]
}
```

Para obter mais informações, consulte [Trabalhando com gateways locais no Guia do Usuário do AWS Outposts](#).

- Para API obter detalhes, consulte [DescribeLocalGatewayVirtualInterfaces](#) na Referência de AWS CLI Comandos.

describe-local-gateways

O código de exemplo a seguir mostra como usar `describe-local-gateways`.

AWS CLI

Para descrever seus gateways locais

O `describe-local-gateways` exemplo a seguir exibe detalhes dos gateways locais que estão disponíveis para você.

```
aws ec2 describe-local-gateways
```

Saída:

```
{
  "LocalGateways": [
    {
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0dc11b66ed59f995a",
      "OwnerId": "123456789012",
      "State": "available"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLocalGateways](#) na Referência de AWS CLI Comandos.

describe-locked-snapshots

O código de exemplo a seguir mostra como usar `describe-locked-snapshots`.

AWS CLI

Para descrever o status de bloqueio de um snapshot

O `describe-locked-snapshots` exemplo a seguir descreve o status de bloqueio do instantâneo especificado.

```
aws ec2 describe-locked-snapshots \
  --snapshot-ids snap-0b5e733b4a8df6e0d
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
    }
  ]
}
```

Para obter mais informações, consulte [Snapshot lock](#) no Guia do EBS usuário da Amazon.

- Para API obter detalhes, consulte [DescribeLockedSnapshots](#) na Referência de AWS CLI Comandos.

describe-managed-prefix-lists

O código de exemplo a seguir mostra como usar `describe-managed-prefix-lists`.

AWS CLI

Para descrever listas de prefixos gerenciadas

O `describe-managed-prefix-lists` exemplo a seguir descreve as listas de prefixos pertencentes à AWS conta `123456789012`.

```
aws ec2 describe-managed-prefix-lists \
  --filters Name=owner-id,Values=123456789012
```

Saída:

```
{
  "PrefixLists": [
    {
      "PrefixListId": "pl-11223344556677aab",
      "AddressFamily": "IPv6",
      "State": "create-complete",
      "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/pl-11223344556677aab",
      "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"
    }
]
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DescribeManagedPrefixLists](#) na Referência de AWS CLI Comandos.

describe-moving-addresses

O código de exemplo a seguir mostra como usar `describe-moving-addresses`.

AWS CLI

Para descrever seus endereços móveis

Este exemplo descreve todos os seus endereços IP elásticos em movimento.

Comando:

```
aws ec2 describe-moving-addresses
```

Saída:

```
{
  "MovingAddressStatuses": [
    {
      "PublicIp": "198.51.100.0",
      "MoveStatus": "MovingToVpc"
    }
  ]
}
```

Este exemplo descreve todos os endereços que estão sendo movidos para a VPC plataforma EC2 -.

Comando:

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- Para API obter detalhes, consulte [DescribeMovingAddresses](#) na Referência de AWS CLI Comandos.

describe-nat-gateways

O código de exemplo a seguir mostra como usar describe-nat-gateways.

AWS CLI

Exemplo 1: Para descrever um NAT gateway público

O describe-nat-gateways exemplo a seguir descreve o NAT gateway público especificado.

```
aws ec2 describe-nat-gateways \  
  --nat-gateway-id nat-01234567890abcdef
```

Saída:

```
{  
  "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"
}
]
}

```

Exemplo 2: Para descrever um NAT gateway privado

O `describe-nat-gateways` exemplo a seguir descreve o NAT gateway privado especificado.

```

aws ec2 describe-nat-gateways \
  --nat-gateway-id nat-1234567890abcdef0

```

Saída:

```

{
  "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"
}
]
}

```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DescribeNatGateways](#) na Referência de AWS CLI Comandos.

describe-network-acls

O código de exemplo a seguir mostra como usar `describe-network-acls`.

AWS CLI

Para descrever sua rede ACLs

O `describe-network-acls` exemplo a seguir recupera detalhes sobre sua redeACLs.

```
aws ec2 describe-network-acls
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Rede ACLs](#) no Guia AWS VPC do Usuário.

- Para API obter detalhes, consulte [DescribeNetworkAcls](#) na Referência de AWS CLI Comandos.

describe-network-insights-access-scope-analyses

O código de exemplo a seguir mostra como usar `describe-network-insights-access-scope-analyses`.

AWS CLI

Para descrever as análises do escopo de acesso do Network Insights

O `describe-network-insights-access-scope-analyses` exemplo a seguir descreve a análise do escopo de acesso em sua AWS conta.

```
aws ec2 describe-network-insights-access-scope-analyses \  
  --region us-east-1
```

Saída:

```
{  
  "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": []  
    }  
  ]  
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI Network Access Analyzer](#).

- Para API obter detalhes, consulte [DescribeNetworkInsightsAccessScopeAnalyses](#) na Referência de AWS CLI Comandos.

`describe-network-insights-access-scopes`

O código de exemplo a seguir mostra como usar `describe-network-insights-access-scopes`.

AWS CLI

Para descrever os escopos de acesso do Network Insights

O `describe-network-insights-access-scopes` exemplo a seguir descreve as análises do escopo de acesso em sua AWS conta.

```
aws ec2 describe-network-insights-access-scopes \  
  --region us-east-1
```

```
--region us-east-1
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI](#) Network Access Analyzer.

- Para API obter detalhes, consulte [DescribeNetworkInsightsAccessScopes](#) na Referência de AWS CLI Comandos.

describe-network-insights-analyses

O código de exemplo a seguir mostra como usar `describe-network-insights-analyses`.

AWS CLI

Para visualizar os resultados de uma análise de caminho

O `describe-network-insights-analyses` exemplo a seguir descreve a análise especificada. Neste exemplo, a origem é um gateway da Internet, o destino é uma EC2 instância e o protocolo é TCP. A análise foi bem-sucedida (`Status` é `succeeded`) e o caminho não é alcançável (`NetworkPathFound` é `false`). O código de explicação `ENI_SG_RULES_MISMATCH` indica que o grupo de segurança da instância não contém uma regra que permita tráfego na porta de destino.

```
aws ec2 describe-network-insights-analyses \
  --network-insights-analysis-ids nia-02207aa13eb480c7a
```

Saída:

```
{
  "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": []
    }
  ]
}
```



```
}
```

Para obter mais informações, consulte [Introdução ao uso do AWS CLI no Guia](#) do Reachability Analyzer.

- Para API obter detalhes, consulte [DescribeNetworkInsightsAnalyses](#) na Referência de AWS CLI Comandos.

describe-network-insights-paths

O código de exemplo a seguir mostra como usar `describe-network-insights-paths`.

AWS CLI

Para descrever um caminho

O `describe-network-insights-paths` exemplo a seguir descreve o caminho especificado.

```
aws ec2 describe-network-insights-paths \  
  --network-insights-path-ids nip-0b26f224f1d131fa8
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Introdução ao uso do AWS CLI no Guia](#) do Reachability Analyzer.

- Para API obter detalhes, consulte [DescribeNetworkInsightsPaths](#) na Referência de AWS CLI Comandos.

describe-network-interface-attribute

O código de exemplo a seguir mostra como usar `describe-network-interface-attribute`.

AWS CLI

Para descrever o atributo de anexo de uma interface de rede

Este exemplo de comando descreve o `attachment` atributo da interface de rede especificada.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute attachment
```

Saída:

```
{  
  "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"  
  }  
}
```

Para descrever o atributo de descrição de uma interface de rede

Este exemplo de comando descreve o `description` atributo da interface de rede especificada.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute description
```

Saída:

```
{
```

```
"NetworkInterfaceId": "eni-686ea200",
  "Description": {
    "Value": "My description"
  }
}
```

Para descrever o `groupSet` atributo de uma interface de rede

Este exemplo de comando descreve o `groupSet` atributo da interface de rede especificada.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute groupSet
```

Saída:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
    {
      "GroupName": "my-security-group",
      "GroupId": "sg-903004f8"
    }
  ]
}
```

Para descrever o `sourceDestCheck` atributo de uma interface de rede

Este exemplo de comando descreve o `sourceDestCheck` atributo da interface de rede especificada.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute sourceDestCheck
```

Saída:

```
{
```

```
"NetworkInterfaceId": "eni-686ea200",
"SourceDestCheck": {
  "Value": true
}
}
```

- Para API obter detalhes, consulte [DescribeNetworkInterfaceAttribute](#) na Referência de AWS CLI Comandos.

describe-network-interface-permissions

O código de exemplo a seguir mostra como usar `describe-network-interface-permissions`.

AWS CLI

Para descrever suas permissões de interface de rede

Este exemplo descreve todas as suas permissões de interface de rede.

Comando:

```
aws ec2 describe-network-interface-permissions
```

Saída:

```
{
  "NetworkInterfacePermissions": [
    {
      "PermissionState": {
        "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-b909511a",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeNetworkInterfacePermissions](#) na Referência de AWS CLI Comandos.

describe-network-interfaces

O código de exemplo a seguir mostra como usar `describe-network-interfaces`.

AWS CLI

Para descrever suas interfaces de rede

Este exemplo descreve todas as suas interfaces de rede.

Comando:

```
aws ec2 describe-network-interfaces
```

Saída:

```
{
  "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"
  }
]
}

```

Este exemplo descreve interfaces de rede que têm uma tag com a chave `Purpose` e o valor `Prod`.

Comando:

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

Saída:

```

{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [DescribeNetworkInterfaces](#) na Referência de AWS CLI Comandos.

describe-placement-groups

O código de exemplo a seguir mostra como usar `describe-placement-groups`.

AWS CLI

Para descrever seus grupos de colocação

Este exemplo de comando descreve todos os seus grupos de posicionamento.

Comando:

```
aws ec2 describe-placement-groups
```

Saída:

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster",
      "State": "available",
      "Strategy": "cluster"
    },
    ...
  ]
}
```

- Para API obter detalhes, consulte [DescribePlacementGroups](#) na Referência de AWS CLI Comandos.

describe-prefix-lists

O código de exemplo a seguir mostra como usar `describe-prefix-lists`.

AWS CLI

Para descrever listas de prefixos

Este exemplo lista todas as listas de prefixos disponíveis para a região.

Comando:

```
aws ec2 describe-prefix-lists
```

Saída:

```
{
  "PrefixLists": [
    {
      "PrefixListName": "com.amazonaws.us-east-1.s3",
      "Cidrs": [
        "54.231.0.0/17"
      ],
      "PrefixListId": "pl-63a5400a"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribePrefixLists](#) na Referência de AWS CLI Comandos.

describe-principal-id-format

O código de exemplo a seguir mostra como usar `describe-principal-id-format`.

AWS CLI

Para descrever o formato de ID para IAM usuários e funções com o formato de ID longo ativado

O `describe-principal-id-format` exemplo a seguir descreve o formato de ID para o usuário raiz, todas as IAM funções e todos os IAM usuários com o formato de ID longo ativado.

```
aws ec2 describe-principal-id-format \
  --resource instance
```

Saída:

```
{
  "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
    },
]
},
...
]
}

```

- Para API obter detalhes, consulte [DescribePrincipalIdFormat](#) na Referência de AWS CLI Comandos.

describe-public-ipv4-pools

O código de exemplo a seguir mostra como usar `describe-public-ipv4-pools`.

AWS CLI

Para descrever seus grupos IPv4 de endereços públicos

O `describe-public-ipv4-pools` exemplo a seguir exibe detalhes sobre os grupos de endereços que foram criados quando você provisionou intervalos de IPv4 endereços públicos usando Bring Your Own IP Addresses (BYOIP).

```
aws ec2 describe-public-ipv4-pools
```

Saída:

```

{
  "PublicIpv4Pools": [
    {
      "PoolId": "ipv4pool-ec2-1234567890abcdef0",
      "PoolAddressRanges": [
        {
          "FirstAddress": "203.0.113.0",
          "LastAddress": "203.0.113.255",
          "AddressCount": 256,

```

```
        "AvailableAddressCount": 256
      }
    ],
    "TotalAddressCount": 256,
    "TotalAvailableAddressCount": 256
  }
]
```

- Para API obter detalhes, consulte [DescribePublicIpv4Pools](#) na Referência de AWS CLI Comandos.

describe-regions

O código de exemplo a seguir mostra como usar `describe-regions`.

AWS CLI

Exemplo 1: descrever todas as suas regiões habilitadas

O exemplo `describe-regions` a seguir descreve todas as regiões habilitadas para a sua conta.

```
aws ec2 describe-regions
```

Saída:

```
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.ap-south-1.amazonaws.com",
      "RegionName": "ap-south-1",
      "OptInStatus": "opt-in-not-required"
    },
    {
      "Endpoint": "ec2.eu-west-3.amazonaws.com",
      "RegionName": "eu-west-3",
```

```
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-west-2.amazonaws.com",
    "RegionName": "eu-west-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-west-1.amazonaws.com",
    "RegionName": "eu-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
    "RegionName": "ap-northeast-3",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-2.amazonaws.com",
    "RegionName": "ap-northeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-northeast-1.amazonaws.com",
    "RegionName": "ap-northeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.sa-east-1.amazonaws.com",
    "RegionName": "sa-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ca-central-1.amazonaws.com",
    "RegionName": "ca-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
    "RegionName": "ap-southeast-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
```

```
    "RegionName": "ap-southeast-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.eu-central-1.amazonaws.com",
    "RegionName": "eu-central-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-1.amazonaws.com",
    "RegionName": "us-east-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-east-2.amazonaws.com",
    "RegionName": "us-east-2",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-1.amazonaws.com",
    "RegionName": "us-west-1",
    "OptInStatus": "opt-in-not-required"
  },
  {
    "Endpoint": "ec2.us-west-2.amazonaws.com",
    "RegionName": "us-west-2",
    "OptInStatus": "opt-in-not-required"
  }
]
}
```

Para obter mais informações, consulte [Regiões e zonas](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: para descrever as regiões habilitadas com um endpoint cujo nome contém uma string específica

O exemplo `describe-regions` a seguir descreve todas as regiões que você habilitou e que têm a string “us” no endpoint.

```
aws ec2 describe-regions \
  --filters "Name=endpoint,Values=*us*"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Regiões e zonas](#) no Guia EC2 do usuário da Amazon.

Exemplo 3: descrever todas as regiões

O exemplo `describe-regions` a seguir descreve todas as regiões disponíveis, incluindo as que estão desabilitadas.

```
aws ec2 describe-regions \
  --all-regions
```

Saída:

```
{
  "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"
  }
}
```

```
]
}
```

Para obter mais informações, consulte [Regiões e zonas](#) no Guia EC2 do usuário da Amazon.

Exemplo 4: listar somente os nomes das regiões

O exemplo `describe-regions` a seguir usa o parâmetro `--query` para filtrar a saída e retornar somente os nomes das regiões como texto.

```
aws ec2 describe-regions \
  --all-regions \
  --query "Regions[].{Name:RegionName}" \
  --output text
```

Saída:

```
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
```

Para obter mais informações, consulte [Regiões e zonas](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeRegions](#) na Referência de AWS CLI Comandos.

describe-replace-root-volume-tasks

O código de exemplo a seguir mostra como usar `describe-replace-root-volume-tasks`.

AWS CLI

Exemplo 1: Para visualizar informações sobre uma tarefa específica de substituição do volume raiz

O `describe-replace-root-volume-tasks` exemplo a seguir descreve a tarefa de substituição do volume raiz `replacevol-0111122223333abcd`.

```
aws ec2 describe-replace-root-volume-tasks \
  --replace-root-volume-task-ids replacevol-0111122223333abcd
```

Saída:

```
{
  "ReplaceRootVolumeTasks": [
    {
      "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
      "Tags": [],
      "InstanceId": "i-0123456789abcdefa",
      "TaskState": "succeeded",
      "StartTime": "2022-03-14T15:16:28Z",
      "CompleteTime": "2022-03-14T15:16:52Z"
    }
  ]
}
```

Para obter mais informações, consulte [Substituir um volume raiz](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Exemplo 2: Para visualizar informações sobre todas as tarefas de substituição do volume raiz de uma instância específica

O `describe-replace-root-volume-tasks` exemplo a seguir descreve todas as tarefas de substituição do volume raiz, por exemplo `i-0123456789abcdefa`.

```
aws ec2 describe-replace-root-volume-tasks \
  --filters Name=instance-id,Values=i-0123456789abcdefa
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Substituir um volume raiz](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [DescribeReplaceRootVolumeTasks](#) na Referência de AWS CLI Comandos.

describe-reserved-instances-listings

O código de exemplo a seguir mostra como usar `describe-reserved-instances-listings`.

AWS CLI

Para descrever uma lista de instâncias reservadas

O `describe-reserved-instances-listings` exemplo a seguir recupera informações sobre a listagem de Instâncias Reservadas especificada.

```
aws ec2 describe-reserved-instances-listings \
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DescribeReservedInstancesListings](#) na Referência de AWS CLI Comandos.

describe-reserved-instances-modifications

O código de exemplo a seguir mostra como usar `describe-reserved-instances-modifications`.

AWS CLI

Para descrever as modificações das instâncias reservadas

Este exemplo de comando descreve todas as solicitações de modificação de instâncias reservadas que foram enviadas para sua conta.

Comando:

```
aws ec2 describe-reserved-instances-modifications
```

Saída:

```
{
  "ReservedInstancesModifications": [
    {
      "Status": "fulfilled",
      "ModificationResults": [
        {
          "ReservedInstancesId": "93bbca2-62f1-4d9d-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"
    }
  ]
}
]
}
}

```

- Para API obter detalhes, consulte [DescribeReservedInstancesModifications](#) na Referência de AWS CLI Comandos.

describe-reserved-instances-offerings

O código de exemplo a seguir mostra como usar `describe-reserved-instances-offerings`.

AWS CLI

Para descrever as ofertas de instâncias reservadas

Este exemplo de comando descreve todas as instâncias reservadas disponíveis para compra na região.

Comando:

```
aws ec2 describe-reserved-instances-offerings
```

Saída:

```
{
  "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"
},
...
}
```

Para descrever suas ofertas de instâncias reservadas usando opções

Este exemplo lista as instâncias reservadas oferecidas pela AWS com as seguintes especificações: tipos de instância t1.micro, produtos Windows VPC (Amazon) e ofertas de utilização pesada.

Comando:

```
aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-type "t1.micro" --product-description "Windows (Amazon VPC)" --offering-type "no upfront"
```

Saída:

```
{
  "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"
  }
]
```

- Para API obter detalhes, consulte [DescribeReservedInstancesOfferings](#) na Referência de AWS CLI Comandos.

describe-reserved-instances

O código de exemplo a seguir mostra como usar `describe-reserved-instances`.

AWS CLI

Para descrever suas instâncias reservadas

Este exemplo de comando descreve as instâncias reservadas que você possui.

Comando:

```
aws ec2 describe-reserved-instances
```

Saída:

```
{
  "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
  },
  ...
]
}

```

Para descrever suas instâncias reservadas usando filtros

Este exemplo filtra a resposta para incluir somente instâncias t2.micro UNIX Linux/Reservadas t2.micro de três anos em 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

```

Saída:

```

{
  "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
  }
]
```

Para obter mais informações, consulte [Como usar EC2 instâncias da Amazon](#) no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [DescribeReservedInstances](#) na Referência de AWS CLI Comandos.

describe-route-tables

O código de exemplo a seguir mostra como usar `describe-route-tables`.

AWS CLI

Para descrever suas tabelas de rotas

O `describe-route-tables` exemplo a seguir recupera os detalhes sobre suas tabelas de rotas

```
aws ec2 describe-route-tables
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Trabalhando com tabelas de rotas](#) no Guia AWS VPC do usuário.

- Para API obter detalhes, consulte [DescribeRouteTables](#) na Referência de AWS CLI Comandos.

describe-scheduled-instance-availability

O código de exemplo a seguir mostra como usar `describe-scheduled-instance-availability`.

AWS CLI

Para descrever um cronograma disponível

Este exemplo descreve uma programação que ocorre toda semana no domingo, começando na data especificada.

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
```

Saída:

```
{  
  "ScheduledInstanceAvailabilitySet": [  
    {  
      "AvailabilityZone": "us-west-2b",  
      "TotalScheduledInstanceHours": 1219,  
      "PurchaseToken": "eyJ2IjoiMSIsInMiOiJEsImMiOi...",  
      "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"
```

```
    },  
    ...  
  ]  
}
```

Para restringir os resultados, você pode adicionar filtros que especificam o sistema operacional, a rede e o tipo de instância.

Comando:

```
--filters nome=plataforma, valores=Linux/ nome=plataforma de rede, valores= - nome=tipo de  
instância, UNIX valores=C4.large EC2 VPC
```

- Para API obter detalhes, consulte [DescribeScheduledInstanceAvailability](#) na Referência de AWS CLI Comandos.

describe-scheduled-instances

O código de exemplo a seguir mostra como usar `describe-scheduled-instances`.

AWS CLI

Para descrever suas instâncias programadas

Este exemplo descreve a instância agendada especificada.

Comando:

```
aws ec2 describe-scheduled-instances --scheduled-instance-  
ids sci-1234-1234-1234-1234-123456789012
```

Saída:

```
{  
  "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"
}
]
}
```

Este exemplo descreve todas as suas instâncias programadas.

Comando:

```
aws ec2 describe-scheduled-instances
```

- Para API obter detalhes, consulte [DescribeScheduledInstances](#) na Referência de AWS CLI Comandos.

describe-security-group-references

O código de exemplo a seguir mostra como usar `describe-security-group-references`.

AWS CLI

Para descrever referências de grupos de segurança

Este exemplo descreve as referências do grupo de segurança `parasg-bbbb2222`. A resposta indica que o grupo de segurança `sg-bbbb2222` está sendo referenciado por um grupo de segurança em VPC `vpc-aaaaaaaa`.

Comando:


```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

Saída:

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeSecurityGroupReferences](#) na Referência de AWS CLI Comandos.

describe-security-group-rules

O código de exemplo a seguir mostra como usar `describe-security-group-rules`.

AWS CLI

Exemplo 1: Para descrever as regras do grupo de segurança para um grupo de segurança

O `describe-security-group-rules` exemplo a seguir descreve as regras do grupo de segurança de um grupo de segurança especificado. Use a `filters` opção para definir o escopo dos resultados para um grupo de segurança específico.

```
aws ec2 describe-security-group-rules \
  --filters Name="group-id",Values="sg-1234567890abcdef0"
```

Saída:

```
{
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-abcdef01234567890",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "111122223333",
      "IsEgress": false,
```

```

    "IpProtocol": "-1",
    "FromPort": -1,
    "ToPort": -1,
    "ReferencedGroupInfo": {
      "GroupId": "sg-1234567890abcdef0",
      "UserId": "111122223333"
    },
    "Tags": []
  },
  {
    "SecurityGroupId": "sgr-bcdef01234567890a",
    "GroupId": "sg-1234567890abcdef0",
    "GroupOwnerId": "111122223333",
    "IsEgress": true,
    "IpProtocol": "-1",
    "FromPort": -1,
    "ToPort": -1,
    "CidrIpv6": "::/0",
    "Tags": []
  },
  {
    "SecurityGroupId": "sgr-cdef01234567890ab",
    "GroupId": "sg-1234567890abcdef0",
    "GroupOwnerId": "111122223333",
    "IsEgress": true,
    "IpProtocol": "-1",
    "FromPort": -1,
    "ToPort": -1,
    "CidrIpv4": "0.0.0.0/0",
    "Tags": []
  }
]
}

```

Exemplo 2: Para descrever uma regra de grupo de segurança

O `describe-security-group-rules` exemplo a seguir descreve a regra de grupo de segurança especificada.

```

aws ec2 describe-security-group-rules \
  --security-group-rule-ids sgr-cdef01234567890ab

```

Saída:

```
{
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-cdef01234567890ab",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "111122223333",
      "IsEgress": true,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv4": "0.0.0.0/0",
      "Tags": []
    }
  ]
}
```

Para obter mais informações, consulte [Regras de grupos de segurança](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeSecurityGroupRules](#) na Referência de AWS CLI Comandos.

describe-security-groups

O código de exemplo a seguir mostra como usar `describe-security-groups`.

AWS CLI

Exemplo 1: descrever um grupo de segurança

O exemplo `describe-security-groups` a seguir descreve o grupo de segurança especificado.

```
aws ec2 describe-security-groups \
  --group-ids sg-903004f8
```

Saída:

```
{
  "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",
    }
]
}
```

Exemplo 2: descrever grupos de segurança com regras específicas

O `describe-security-groups` exemplo a seguir usa filtros para definir o escopo dos resultados para grupos de segurança que têm uma regra que permite SSH tráfego (porta 22) e uma regra que permite tráfego de todos os endereços (`0.0.0.0/0`). O exemplo usa o parâmetro `--query` para exibir somente os nomes dos grupos de segurança. Os grupos de segurança devem corresponder a todos os filtros para serem retornados nos resultados; no entanto, uma única regra não precisa corresponder a todos os filtros. Por exemplo, a saída retorna um grupo de segurança com uma regra que permite o SSH tráfego de um endereço IP específico e outra regra que permite o HTTP tráfego de todos os endereços.

```
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
```

Saída:

```
default
my-security-group
web-servers
launch-wizard-1
```

Exemplo 3: descrever grupos de segurança com base em tags

O exemplo a seguir `describe-security-groups` usa filtros para definir o escopo dos resultados para grupos de segurança que incluem `test` no nome do grupo de segurança e contêm a tag `Test=To-delete`. O exemplo usa o `--query` parâmetro para exibir somente os nomes e os grupos IDs de segurança.

```
aws ec2 describe-security-groups \
```

```
--filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \  
--query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

Saída:

```
[  
  {  
    "Name": "testfornewinstance",  
    "ID": "sg-33bb22aa"  
  },  
  {  
    "Name": "newgroupptest",  
    "ID": "sg-1a2b3c4d"  
  }  
]
```

Para obter mais exemplos de uso de filtros de tags, consulte Como [trabalhar com tags](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeSecurityGroups](#) na Referência de AWS CLI Comandos.

describe-snapshot-attribute

O código de exemplo a seguir mostra como usar `describe-snapshot-attribute`.

AWS CLI

Para descrever os atributos de um instantâneo

O `describe-snapshot-attribute` exemplo a seguir lista as contas com as quais um snapshot é compartilhado.

```
aws ec2 describe-snapshot-attribute \  
--snapshot-id snap-01234567890abcdef \  
--attribute createVolumePermission
```

Saída:

```
{  
  "SnapshotId": "snap-01234567890abcdef",
```

```
"CreateVolumePermissions": [  
  {  
    "UserId": "123456789012"  
  }  
]  
}
```

Para obter mais informações, consulte [Compartilhe um EBS snapshot da Amazon no Guia](#) do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [DescribeSnapshotAttribute](#) na Referência de AWS CLI Comandos.

describe-snapshot-tier-status

O código de exemplo a seguir mostra como usar `describe-snapshot-tier-status`.

AWS CLI

Para visualizar informações de arquivamento sobre um instantâneo arquivado

O `describe-snapshot-tier-status` exemplo a seguir fornece informações de arquivamento sobre um instantâneo arquivado.

```
aws ec2 describe-snapshot-tier-status \  
  --filters "Name=snapshot-id, Values=snap-01234567890abcdef"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [View archived snapshots](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [DescribeSnapshotTierStatus](#) na Referência de AWS CLI Comandos.

describe-snapshots

O código de exemplo a seguir mostra como usar `describe-snapshots`.

AWS CLI

Exemplo 1: descrever um snapshot

O exemplo `describe-snapshots` a seguir descreve o snapshot especificado.

```
aws ec2 describe-snapshots \  
  --snapshot-ids snap-1234567890abcdef0
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```



```

    ]
  }
]
}

```

Para obter mais informações, consulte [Amazon EBS snapshots](#) no Guia do EC2 usuário da Amazon.

Exemplo 2: descrever snapshots com base em filtros

O `describe-snapshots` exemplo a seguir usa filtros para definir o escopo dos resultados para instantâneos pertencentes à sua AWS conta que estão no `pending` estado. O exemplo usa o `--query` parâmetro para exibir somente o instantâneo IDs e a hora em que o instantâneo foi iniciado.

```

aws ec2 describe-snapshots \
  --owner-ids self \
  --filters Name=status,Values=pending \
  --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"

```

Saída:

```

[
  {
    "ID": "snap-1234567890abcdef0",
    "Time": "2019-08-04T12:48:18.000Z"
  },
  {
    "ID": "snap-066877671789bd71b",
    "Time": "2019-08-04T02:45:16.000Z"
  },
  ...
]

```

O exemplo `describe-snapshots` a seguir usa filtros para definir o escopo dos resultados para snapshots criados no volume especificado. O exemplo usa o `--query` parâmetro para exibir somente o `instantâneoIDs`.

```

aws ec2 describe-snapshots \
  --filters Name=volume-id,Values=049df61146c4d7901 \

```

```
--query "Snapshots[*].[SnapshotId]" \  
--output text
```

Saída:

```
snap-1234567890abcdef0  
snap-08637175a712c3fb9  
...
```

Para obter exemplos adicionais de uso de filtros, consulte [Listar e filtrar seus recursos](#) no Guia do EC2 usuário da Amazon.

Exemplo 3: descrever snapshots com base em tags

O exemplo `describe-snapshots` a seguir usa filtros de tag para definir o escopo dos resultados para snapshots que tenham a tag `Stack=Prod`.

```
aws ec2 describe-snapshots \  
--filters Name=tag:Stack,Values=prod
```

Para obter um exemplo da saída de `describe-snapshots`, consulte o Exemplo 1.

Para obter mais exemplos de uso de filtros de tags, consulte Como [trabalhar com tags](#) no Guia EC2 do usuário da Amazon.

Exemplo 4: descrever snapshots com base na idade

O `describe-snapshots` exemplo a seguir usa JMESPath expressões para descrever todos os instantâneos criados pela sua AWS conta antes da data especificada. Ele exibe somente o `instantâneoIDs`.

```
aws ec2 describe-snapshots \  
--owner-ids 012345678910 \  
--query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

Para obter exemplos adicionais de uso de filtros, consulte [Listar e filtrar seus recursos](#) no Guia do EC2 usuário da Amazon.

Exemplo 5: visualizar somente snapshots arquivados

O exemplo `describe-snapshots` a seguir lista apenas os snapshots armazenados no nível de arquivamento.

```
aws ec2 describe-snapshots \
  --filters "Name=storage-tier,Values=archive"
```

Saída:

```
{
  "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": []
    },
  ]
}
```

Para obter mais informações, consulte [View archived snapshots](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [DescribeSnapshots](#) na Referência de AWS CLI Comandos.

describe-spot-datafeed-subscription

O código de exemplo a seguir mostra como usar `describe-spot-datafeed-subscription`.

AWS CLI

Para descrever a assinatura do feed de dados da Spot Instance para uma conta

Este exemplo de comando descreve o feed de dados da conta.

Comando:

```
aws ec2 describe-spot-datafeed-subscription
```

Saída:

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "123456789012",
    "Prefix": "spotdata",
    "Bucket": "my-s3-bucket",
    "State": "Active"
  }
}
```

- Para API obter detalhes, consulte [DescribeSpotDatafeedSubscription](#) na Referência de AWS CLI Comandos.

describe-spot-fleet-instances

O código de exemplo a seguir mostra como usar `describe-spot-fleet-instances`.

AWS CLI

Para descrever as instâncias spot associadas a uma frota spot

Este exemplo de comando lista as instâncias spot associadas à frota spot especificada.

Comando:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Saída:

```
{
  "ActiveInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "InstanceType": "m3.medium",
      "SpotInstanceRequestId": "sir-08b93456"
    },
    ...
  ]
}
```

```
],  
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

- Para API obter detalhes, consulte [DescribeSpotFleetInstances](#) na Referência de AWS CLI Comandos.

describe-spot-fleet-request-history

O código de exemplo a seguir mostra como usar `describe-spot-fleet-request-history`.

AWS CLI

Para descrever o histórico da frota Spot

Esse exemplo de comando retorna o histórico da frota spot especificada a partir do horário especificado.

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
```

O exemplo de saída a seguir mostra os lançamentos bem-sucedidos de duas instâncias spot para a frota spot.

Saída:

```
{  
  "HistoryRecords": [  
    {  
      "Timestamp": "2015-05-26T23:17:20.697Z",  
      "EventInformation": {  
        "EventSubType": "submitted"  
      },  
      "EventType": "fleetRequestChange"  
    },  
    {  
      "Timestamp": "2015-05-26T23:17:20.873Z",  
      "EventInformation": {  
        "EventSubType": "active"  
      },  
    },  
  ],  
}
```

```

    "EventType": "fleetRequestChange"
  },
  {
    "Timestamp": "2015-05-26T23:21:21.712Z",
    "EventInformation": {
      "InstanceId": "i-1234567890abcdef0",
      "EventSubType": "launched"
    },
    "EventType": "instanceChange"
  },
  {
    "Timestamp": "2015-05-26T23:21:21.816Z",
    "EventInformation": {
      "InstanceId": "i-1234567890abcdef1",
      "EventSubType": "launched"
    },
    "EventType": "instanceChange"
  }
],
"SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
"NextToken": "CpHNsscimcV5oH7bSbub03CI2Qms5+ypNpNm
+53MNLr0YcXAkp0xFlfKf91yVxSExmbtma3awYxMFzNA663ZskT0AHtJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauVg
+P+TfG1WxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
"StartTime": "2015-05-26T00:00:00Z"
}

```

- Para API obter detalhes, consulte [DescribeSpotFleetRequestHistory](#) na Referência de AWS CLI Comandos.

describe-spot-fleet-requests

O código de exemplo a seguir mostra como usar `describe-spot-fleet-requests`.

AWS CLI

Para descrever suas solicitações de frota Spot

Este exemplo descreve todas as suas solicitações de frota spot.

Comando:

```
aws ec2 describe-spot-fleet-requests
```

Saída:

```
{
  "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"
  }
]
}

```

Para descrever uma solicitação de frota spot

Este exemplo descreve a solicitação de frota spot especificada.

Comando:

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Saída:

```

{
  "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"
}
]
```

- Para API obter detalhes, consulte [DescribeSpotFleetRequests](#) na Referência de AWS CLI Comandos.

describe-spot-instance-requests

O código de exemplo a seguir mostra como usar `describe-spot-instance-requests`.

AWS CLI

Exemplo 1: Para descrever uma solicitação de instância spot

O `describe-spot-instance-requests` exemplo a seguir descreve a solicitação de instância spot especificada.

```
aws ec2 describe-spot-instance-requests \
  --spot-instance-request-ids sir-08b93456
```

Saída:

```
{
  "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"
}
]
}

```

Exemplo 2: Para descrever solicitações de Instância Spot com base em filtros

O `describe-spot-instance-requests` exemplo a seguir usa filtros para definir o escopo dos resultados para solicitações de instância spot com o tipo de instância especificado na zona de disponibilidade especificada. O exemplo usa o `--query` parâmetro para exibir somente a instânciasIDs.

```

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
```

Saída:

```
i-057750d42936e468a  
i-001efd250faaa6ffa  
i-027552a73f021f3bd  
...
```

Para obter exemplos adicionais de uso de filtros, consulte [Listar e filtrar seus recursos no Guia](#) do usuário do Amazon Elastic Compute Cloud.

Exemplo 3: Para descrever solicitações de Instância Spot com base em tags

O `describe-spot-instance-requests` exemplo a seguir usa filtros de tag para definir o escopo dos resultados para solicitações de Instância Spot que têm a `tagcost-center=cc123`.

```
aws ec2 describe-spot-instance-requests \  
  --filters Name=tag:cost-center,Values=cc123
```

Para obter um exemplo da saída de `describe-spot-instance-requests`, consulte o Exemplo 1.

Para obter mais exemplos de uso de filtros de tags, consulte Como [trabalhar com tags](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeSpotInstanceRequests](#) na Referência de AWS CLI Comandos.

describe-spot-price-history

O código de exemplo a seguir mostra como usar `describe-spot-price-history`.

AWS CLI

Para descrever o histórico de preços à vista

Esse exemplo de comando retorna o histórico de preços spot para instâncias `m1.xlarge` em um determinado dia de janeiro.

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
```

Saída:

```
{
  "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"
    },
    ...
  ]
}
```

Para descrever o histórico de preços spot para Linux/Amazon UNIX VPC

Esse comando de exemplo retorna o histórico de preços spot para VPC instâncias m1.xlarge, UNIX Linux/Amazon de um determinado dia de janeiro.

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
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeSpotPriceHistory](#) na Referência de AWS CLI Comandos.

describe-stale-security-groups

O código de exemplo a seguir mostra como usar `describe-stale-security-groups`.

AWS CLI

Para descrever grupos de segurança obsoletos

Este exemplo descreve regras obsoletas do grupo de segurança para `vpc-11223344`. A resposta mostra que `sg-5fa68d3a` em sua conta tem uma regra de entrada obsoleta que faz referência `sg-279ab042` no par VPC e que em sua conta tem uma SSH regra de saída obsoleta que faz referência `sg-fe6fba9a` no par. SSH `sg-ef6fba8b` VPC

Comando:

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

Saída:

```
{
  "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"  
  }  
]  
}
```

- Para API obter detalhes, consulte [DescribeStaleSecurityGroups](#) na Referência de AWS CLI Comandos.

describe-store-image-tasks

O código de exemplo a seguir mostra como usar `describe-store-image-tasks`.

AWS CLI

Para descrever o progresso de uma tarefa da AMI loja

O `describe-store-image-tasks` exemplo a seguir descreve o progresso de uma tarefa de AMI armazenamento.

```
aws ec2 describe-store-image-tasks
```

Saída:

```
{  
  "AmiId": "ami-1234567890abcdef0",  
  "Bucket": "my-ami-bucket",  
  "ProgressPercentage": 17,  
  "S3ObjectKey": "ami-1234567890abcdef0.bin",  
  "StoreTaskState": "InProgress",  
  "StoreTaskFailureReason": null,  
  "TaskStartTime": "2022-01-01T01:01:01.001Z"  
}
```

Para obter mais informações sobre como armazenar, restaurar e AMI usar o S3, consulte [Armazenar e restaurar e AMI usar o S3 <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuideami-store-restore.html>](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuideami-store-restore.html) no Guia do usuário da Amazon. EC2

- Para API obter detalhes, consulte [DescribeStoreImageTasks](#) na Referência de AWS CLI Comandos.

describe-subnets

O código de exemplo a seguir mostra como usar `describe-subnets`.

AWS CLI

Exemplo 1: descrever todas as suas sub-redes

O exemplo `describe-subnets` a seguir mostra os detalhes das suas sub-redes.

```
aws ec2 describe-subnets
```

Saída:

```
{
  "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": "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
    }
  }
]
}

```

Para obter mais informações, consulte [Trabalho com sub-redes VPCs e sub-redes](#) no Guia do AWS VPC usuário.

Exemplo 2: Para descrever as sub-redes de uma rede específica VPC

O `describe-subnets` exemplo a seguir usa um filtro para recuperar detalhes das sub-redes especificadas. VPC

```

aws ec2 describe-subnets \
  --filters "Name=vpc-id,Values=vpc-3EXAMPLE"

```

Saída:

```

{
  "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
      }
    }
  ]
}

```

Para obter mais informações, consulte [Trabalho com sub-redes VPCs e sub-redes](#) no Guia do AWS VPC usuário.

Exemplo 3: descrever as sub-redes com uma tag específica

O `describe-subnets` exemplo a seguir usa um filtro para recuperar os detalhes dessas sub-redes com a tag `CostCenter=123` e o `--query` parâmetro para exibir a sub-rede IDs das sub-redes com essa tag.

```
aws ec2 describe-subnets \  
  --filters "Name=tag:CostCenter,Values=123" \  
  --query "Subnets[*].SubnetId" \  
  --output text
```

Saída:

```
subnet-0987a87c8b37348ef  
subnet-02a95061c45f372ee  
subnet-03f720e7de2788d73
```

Para obter mais informações, consulte [Trabalho com sub-redes VPCs e sub-redes no Guia VPC](#) do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeSubnets](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar describe-tags.

AWS CLI

Exemplo 1: Para descrever todas as tags de um único recurso

O describe-tags exemplo a seguir descreve as tags da instância especificada.

```
aws ec2 describe-tags \  
  --filters "Name=resource-id,Values=i-1234567890abcdef8"
```

Saída:

```
{  
  "Tags": [  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",  
      "Value": "Test",  
      "Key": "Stack"  
    },  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",
```

```
        "Value": "Beta Server",
        "Key": "Name"
    }
]
}
```

Exemplo 2: Para descrever todas as tags de um tipo de recurso

O `describe-tags` exemplo a seguir descreve as tags dos seus volumes.

```
aws ec2 describe-tags \
  --filters "Name=resource-type,Values=volume"
```

Saída:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-1234567890abcdef0",
      "Value": "Project1",
      "Key": "Purpose"
    },
    {
      "ResourceType": "volume",
      "ResourceId": "vol-049df61146c4d7901",
      "Value": "Logs",
      "Key": "Purpose"
    }
  ]
}
```

Exemplo 3: Para descrever todas as suas tags

O `describe-tags` exemplo a seguir descreve as tags de todos os seus recursos.

```
aws ec2 describe-tags
```

Exemplo 4: Para descrever as tags de seus recursos com base em uma chave de tag

O `describe-tags` exemplo a seguir descreve as tags dos seus recursos que têm uma tag com a chave `Stack`.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Stack
```

Saída:

```
{  
  "Tags": [  
    {  
      "ResourceType": "volume",  
      "ResourceId": "vol-027552a73f021f3b",  
      "Value": "Production",  
      "Key": "Stack"  
    },  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",  
      "Value": "Test",  
      "Key": "Stack"  
    }  
  ]  
}
```

Exemplo 5: Para descrever as tags de seus recursos com base na chave e no valor da tag

O describe-tags exemplo a seguir descreve as tags dos seus recursos que têm a tagStack=Test.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Stack Name=value,Values=Test
```

Saída:

```
{  
  "Tags": [  
    {  
      "ResourceType": "image",  
      "ResourceId": "ami-3ac336533f021f3bd",  
      "Value": "Test",  
      "Key": "Stack"  
    },  
    {
```

```

    "ResourceType": "instance",
    "ResourceId": "i-1234567890abcdef8",
    "Value": "Test",
    "Key": "Stack"
  }
]
}

```

O `describe-tags` exemplo a seguir usa uma sintaxe alternativa para descrever recursos com a `tagStack=Test`.

```

aws ec2 describe-tags \
  --filters "Name=tag:Stack,Values=Test"

```

O `describe-tags` exemplo a seguir descreve as tags de todas as suas instâncias que têm uma tag com a chave `Purpose` e sem valor.

```

aws ec2 describe-tags \
  --filters "Name=resource-
  type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="

```

Saída:

```

{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef5",
      "Value": null,
      "Key": "Purpose"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

describe-traffic-mirror-filters

O código de exemplo a seguir mostra como usar `describe-traffic-mirror-filters`.

AWS CLI

Para ver seus filtros de espelhos de trânsito

O `describe-traffic-mirror-filters` exemplo a seguir exibe detalhes de todos os seus filtros de espelhos de tráfego.

```
aws ec2 describe-traffic-mirror-filters
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Exibir seus filtros de espelhamento de tráfego](#) no Guia de espelhamento de tráfego.

- Para API obter detalhes, consulte [DescribeTrafficMirrorFilters](#) na Referência de AWS CLI Comandos.

describe-traffic-mirror-sessions

O código de exemplo a seguir mostra como usar `describe-traffic-mirror-sessions`.

AWS CLI

Para descrever uma sessão de espelhamento de tráfego

O `describe-traffic-mirror-sessions` exemplo a seguir exibe detalhes das suas sessões do Traffic Mirror.

```
aws ec2 describe-traffic-mirror-sessions
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Exibir detalhes da sessão do Traffic Mirror](#) no AWS Traffic Mirroring Guide.

- Para API obter detalhes, consulte [DescribeTrafficMirrorSessions](#) na Referência de AWS CLI Comandos.

describe-traffic-mirror-targets

O código de exemplo a seguir mostra como usar `describe-traffic-mirror-targets`.

AWS CLI

Para descrever um alvo de espelhos de trânsito

O `describe-traffic-mirror-targets` exemplo a seguir exibe informações sobre o alvo do espelhamento de tráfego especificado.

```

aws ec2 describe-traffic-mirror-targets \
  --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE

```

Saída:

```

{
  "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": []
    }
  ]
}

```

Para obter mais informações, consulte [Alvos do espelho de tráfego](#) no Amazon VPC Traffic Mirroring Guide.

- Para API obter detalhes, consulte [DescribeTrafficMirrorTargets](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-attachments

O código de exemplo a seguir mostra como usar `describe-transit-gateway-attachments`.

AWS CLI

Para visualizar seus anexos do Transit Gateway

O `describe-transit-gateway-attachments` exemplo a seguir exibe detalhes dos anexos do gateway de trânsito.

```
aws ec2 describe-transit-gateway-attachments
```

Saída:

```
{
  "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": []
}
```

```

    }
  ]
}

```

Para obter mais informações, consulte [Trabalhar com gateways de transporte](#) público no Transit Gateways Guide.

- Para API obter detalhes, consulte [DescribeTransitGatewayAttachments](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-connect-peers

O código de exemplo a seguir mostra como usar `describe-transit-gateway-connect-peers`.

AWS CLI

Para descrever um peer do Transit Gateway Connect

O `describe-transit-gateway-connect-peers` exemplo a seguir descreve o Connect peer especificado.

```

aws ec2 describe-transit-gateway-connect-peers \
  --transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE

```

Saída:

```

{
  "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": []
}
]
}

```

Para obter mais informações, consulte os [anexos do Transit Gateway Connect e os pares do Transit Gateway Connect](#) no Guia do Transit Gateways.

- Para API obter detalhes, consulte [DescribeTransitGatewayConnectPeers](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-connects

O código de exemplo a seguir mostra como usar `describe-transit-gateway-connects`.

AWS CLI

Para descrever um anexo do Transit Gateway Connect

O `describe-transit-gateway-connects` exemplo a seguir descreve o anexo Connect especificado.

```
aws ec2 describe-transit-gateway-connects \
  --transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte os [anexos do Transit Gateway Connect e os pares do Transit Gateway Connect](#) no Guia do Transit Gateways.

- Para API obter detalhes, consulte [DescribeTransitGatewayConnects](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-multicast-domains

O código de exemplo a seguir mostra como usar `describe-transit-gateway-multicast-domains`.

AWS CLI

Para descrever seus domínios multicast do Transit Gateway

O `describe-transit-gateway-multicast-domains` exemplo a seguir exibe detalhes de todos os seus domínios multicast do Transit Gateway.

```
aws ec2 describe-transit-gateway-multicast-domains
```

Saída:

```
{
  "TransitGatewayMulticastDomains": [
    {
```

```

    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
    "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"
      }
    ]
  }
]
}

```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DescribeTransitGatewayMulticastDomains](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-peering-attachments

O código de exemplo a seguir mostra como usar `describe-transit-gateway-peering-attachments`.

AWS CLI

Para descrever seus anexos de emparelhamento do Transit Gateway

O `describe-transit-gateway-peering-attachments` exemplo a seguir exibe detalhes de todos os anexos de emparelhamento do Transit Gateway.

```
aws ec2 describe-transit-gateway-peering-attachments
```


Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Transit Gateway Peering Attachments](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DescribeTransitGatewayPeeringAttachments](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-policy-tables

O código de exemplo a seguir mostra como usar `describe-transit-gateway-policy-tables`.

AWS CLI

Para descrever uma tabela de políticas de gateway de trânsito

O `describe-transit-gateway-policy-tables` exemplo a seguir descreve a tabela de políticas do gateway de trânsito especificada.

```
aws ec2 describe-transit-gateway-policy-tables \
  --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

Saída:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",
      "State": "available",
      "CreationTime": "2023-11-28T16:36:43+00:00",
      "Tags": []
    }
  ]
}
```

Para obter mais informações, consulte as [tabelas de políticas do Transit Gateway](#) no Guia do usuário do Transit Gateway.

- Para API obter detalhes, consulte [DescribeTransitGatewayPolicyTables](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-route-tables

O código de exemplo a seguir mostra como usar `describe-transit-gateway-route-tables`.

AWS CLI

Para descrever suas tabelas de rotas do Transit Gateway

O `describe-transit-gateway-route-tables` exemplo a seguir exibe detalhes das tabelas de rotas do seu gateway de trânsito.

```
aws ec2 describe-transit-gateway-route-tables
```

Saída:

```
{
  "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": []
    }
]
}

```

Para obter mais informações, consulte [Exibir tabelas de rotas do Transit Gateways](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DescribeTransitGatewayRouteTables](#) na Referência de AWS CLI Comandos.

describe-transit-gateway-vpc-attachments

O código de exemplo a seguir mostra como usar `describe-transit-gateway-vpc-attachments`.

AWS CLI

Para descrever seus VPC anexos do Transit Gateway

O `describe-transit-gateway-vpc-attachments` exemplo a seguir exibe detalhes dos VPC anexos do gateway de trânsito.

```
aws ec2 describe-transit-gateway-vpc-attachments
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}

```

Para obter mais informações, consulte [Exibir seus VPC anexos](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DescribeTransitGatewayVpcAttachments](#) na Referência de AWS CLI Comandos.

describe-transit-gateways

O código de exemplo a seguir mostra como usar `describe-transit-gateways`.

AWS CLI

Para descrever seus portais de trânsito

O `describe-transit-gateways` exemplo a seguir recupera detalhes sobre seus gateways de transporte público.

```
aws ec2 describe-transit-gateways
```

Saída:

```
{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [DescribeTransitGateways](#) na Referência de AWS CLI Comandos.

describe-verified-access-endpoints

O código de exemplo a seguir mostra como usar `describe-verified-access-endpoints`.

AWS CLI

Para descrever um endpoint de acesso verificado

O `delete-verified-access-endpoints` exemplo a seguir descreve o endpoint de acesso verificado especificado.

```

aws ec2 describe-verified-access-endpoints \
  --verified-access-endpoint-ids vae-066fac616d4d546f2

```

Saída:

```

{
  "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"
        }
    ]
}
]
}

```

Para obter mais informações, consulte [Endpoints de acesso verificado](#) no Guia do usuário do AWS Verified Access.

- Para API obter detalhes, consulte [DescribeVerifiedAccessEndpoints](#) na Referência de AWS CLI Comandos.

describe-verified-access-groups

O código de exemplo a seguir mostra como usar `describe-verified-access-groups`.

AWS CLI

Para descrever um grupo de acesso verificado

O `describe-verified-access-groups` exemplo a seguir descreve o grupo de acesso verificado especificado.

```
aws ec2 describe-verified-access-groups \
  --verified-access-group-ids vagr-0dbe967baf14b7235
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Grupos de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DescribeVerifiedAccessGroups](#) na Referência de AWS CLI Comandos.

describe-verified-access-instance-logging-configurations

O código de exemplo a seguir mostra como usar `describe-verified-access-instance-logging-configurations`.

AWS CLI

Para descrever a configuração de registro para uma instância de acesso verificado

O `describe-verified-access-instance-logging-configurations` exemplo a seguir descreve a configuração de registro para a instância de acesso verificado especificada.

```
aws ec2 describe-verified-access-instance-logging-configurations \
```



```
--verified-access-instance-ids vai-0ce000c0b7643abea
```

Saída:

```
{
  "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
      }
    }
  ]
}
```

Para obter mais informações, consulte [Registros de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DescribeVerifiedAccessInstanceLoggingConfigurations](#) na Referência de AWS CLI Comandos.

describe-verified-access-instances

O código de exemplo a seguir mostra como usar `describe-verified-access-instances`.

AWS CLI

Para descrever uma instância de acesso verificado

O `describe-verified-access-instances` exemplo a seguir descreve a instância de acesso verificado especificada.

```
aws ec2 describe-verified-access-instances \
  --verified-access-instance-ids vai-0ce000c0b7643abea
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Instâncias de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DescribeVerifiedAccessInstances](#) na Referência de AWS CLI Comandos.

`describe-verified-access-trust-providers`

O código de exemplo a seguir mostra como usar `describe-verified-access-trust-providers`.

AWS CLI

Para descrever um provedor confiável de acesso verificado

O `describe-verified-access-trust-providers` exemplo a seguir descreve o provedor confiável de acesso verificado especificado.

```
aws ec2 describe-verified-access-trust-providers \
  --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Provedores confiáveis para acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DescribeVerifiedAccessTrustProviders](#) na Referência de AWS CLI Comandos.

describe-volume-attribute

O código de exemplo a seguir mostra como usar `describe-volume-attribute`.

AWS CLI

Para descrever um atributo de volume

Este exemplo de comando descreve o `autoEnableIo` atributo do volume com o ID `vol-049df61146c4d7901`.

Comando:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --  
attribute autoEnableIO
```

Saída:

```
{  
  "AutoEnableIO": {  
    "Value": false  
  },  
  "VolumeId": "vol-049df61146c4d7901"  
}
```

- Para API obter detalhes, consulte [DescribeVolumeAttribute](#) na Referência de AWS CLI Comandos.

describe-volume-status

O código de exemplo a seguir mostra como usar `describe-volume-status`.

AWS CLI

Para descrever o status de um único volume

Este exemplo de comando descreve o status do volume `vol-1234567890abcdef0`.

Comando:

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

Saída:

```
{  
  "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": []
}
]
```

Para descrever o status dos volumes danificados

Este exemplo de comando descreve o status de todos os volumes que estão danificados. Neste exemplo de saída, não há volumes comprometidos.

Comando:

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

Saída:

```
{
  "VolumeStatuses": []
}
```

Se você tiver um volume com falha na verificação de status (o status está comprometido), consulte Como trabalhar com um volume prejudicado no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeVolumeStatus](#) na Referência de AWS CLI Comandos.

describe-volumes-modifications

O código de exemplo a seguir mostra como usar `describe-volumes-modifications`.

AWS CLI

Para descrever o status de modificação de um volume

O `describe-volumes-modifications` exemplo a seguir descreve o status de modificação do volume especificado.

```
aws ec2 describe-volumes-modifications \  
  --volume-ids vol-1234567890abcdef0
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [DescribeVolumesModifications](#) na Referência de AWS CLI Comandos.

describe-volumes

O código de exemplo a seguir mostra como usar `describe-volumes`.

AWS CLI

Exemplo 1: Para descrever um volume

O `describe-volumes` exemplo a seguir descreve os volumes especificados na região atual.

```
aws ec2 describe-volumes \  
  --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

```

    }
  ]
}

```

Exemplo 2: Para descrever volumes anexados a uma instância específica

O `describe-volumes` exemplo a seguir descreve todos os volumes anexados à instância especificada e definidos para serem excluídos quando a instância for encerrada.

```

aws ec2 describe-volumes \
  --region us-east-1 \
  --filters Name=attachment.instance-
id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true

```

Para obter um exemplo da saída de `describe-volumes`, consulte o Exemplo 1.

Exemplo 3: Para descrever os volumes disponíveis em uma zona de disponibilidade específica

O `describe-volumes` exemplo a seguir descreve todos os volumes que têm um status de `available` e estão na zona de disponibilidade especificada.

```

aws ec2 describe-volumes \
  --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a

```

Para obter um exemplo da saída de `describe-volumes`, consulte o Exemplo 1.

Exemplo 4: Para descrever volumes com base em tags

O `describe-volumes` exemplo a seguir descreve todos os volumes que têm a chave de tag `Name` e um valor que começa com `Test`. A saída é então filtrada com uma consulta que exibe somente as tags e IDs os volumes.

```

aws ec2 describe-volumes \
  --filters Name=tag:Name,Values=Test* \
  --query "Volumes[*].{ID:VolumeId,Tag:Tags}"

```

Saída:

```

[
  {

```



```
    "Tag": [
      {
        "Value": "Test2",
        "Key": "Name"
      }
    ],
    "ID": "vol-1234567890abcdef0"
  },
  {
    "Tag": [
      {
        "Value": "Test1",
        "Key": "Name"
      }
    ],
    "ID": "vol-049df61146c4d7901"
  }
]
```

Para obter mais exemplos de uso de filtros de tags, consulte Como [trabalhar com tags](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeVolumes](#) na Referência de AWS CLI Comandos.

describe-vpc-attribute

O código de exemplo a seguir mostra como usar `describe-vpc-attribute`.

AWS CLI

Para descrever o `enableDnsSupport` atributo

Este exemplo descreve o `enableDnsSupport` atributo. Esse atributo indica se a DNS resolução está habilitada para VPC o. Se esse atributo for `true`, o DNS servidor da Amazon resolverá os DNS nomes de host de suas instâncias para os endereços IP correspondentes; caso contrário, isso não acontece.

Comando:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

Saída:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsSupport": {
    "Value": true
  }
}
```

Para descrever o `enableDnsHostnames` atributo

Este exemplo descreve o `enableDnsHostnames` atributo. Esse atributo indica se as instâncias foram executadas no VPC get DNS hostnames. Se esse atributo for `true`, instâncias em VPC get DNS hostnames; caso contrário, não o farão.

Comando:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

Saída:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsHostnames": {
    "Value": true
  }
}
```

- Para API obter detalhes, consulte [DescribeVpcAttribute](#) na Referência de AWS CLI Comandos.

describe-vpc-classic-link-dns-support

O código de exemplo a seguir mostra como usar `describe-vpc-classic-link-dns-support`.

AWS CLI

Para descrever ClassicLink DNS o suporte para seu VPCs

Este exemplo descreve o status de ClassicLink DNS suporte de todos os seus VPCs.

Comando:

```
aws ec2 describe-vpc-classic-link-dns-support
```

Saída:

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeVpcClassicLinkDnsSupport](#) na Referência de AWS CLI Comandos.

describe-vpc-classic-link

O código de exemplo a seguir mostra como usar `describe-vpc-classic-link`.

AWS CLI

Para descrever o ClassicLink status do seu VPCs

Este exemplo lista o ClassicLink status de `vpc-88888888`.

Comando:

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

Saída:

```
{
  "Vpcs": [
    {
      "ClassicLinkEnabled": true,
      "VpcId": "vpc-88888888",
      "Tags": [
        {
```

```
        "Value": "classiclinkvpc",
        "Key": "Name"
      }
    ]
  }
]
```

Este exemplo lista somente os VPCs que estão habilitados para o Classiclink (o valor do filtro de `is-classic-link-enabled` está definido como). `true`

Comando:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-enabled,Values=true"
```

- Para API obter detalhes, consulte [DescribeVpcClassicLink](#) na Referência de AWS CLI Comandos.

describe-vpc-endpoint-connection-notifications

O código de exemplo a seguir mostra como usar `describe-vpc-endpoint-connection-notifications`.

AWS CLI

Para descrever as notificações de conexão do endpoint

O `describe-vpc-endpoint-connection-notifications` exemplo a seguir descreve todas as notificações de conexão do endpoint.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [DescribeVpcEndpointConnectionNotifications](#) na Referência de AWS CLI Comandos.

describe-vpc-endpoint-connections

O código de exemplo a seguir mostra como usar `describe-vpc-endpoint-connections`.

AWS CLI

Para descrever conexões VPC de endpoint

Este exemplo descreve as conexões do endpoint da interface com seu serviço de endpoint e filtra os resultados para exibir os endpoints que estão. `PendingAcceptance`

Comando:

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

Saída:

```

{
  "VpcEndpointConnections": [
    {
      "VpcEndpointId": "vpce-0abed31004e618123",
      "ServiceId": "vpce-svc-0abced088d20def56",
      "CreationTimestamp": "2017-11-30T10:00:24.350Z",
      "VpcEndpointState": "pendingAcceptance",

```

```
    "VpcEndpointOwner": "123456789012"
  }
]
}
```

- Para API obter detalhes, consulte [DescribeVpcEndpointConnections](#) na Referência de AWS CLI Comandos.

describe-vpc-endpoint-service-configurations

O código de exemplo a seguir mostra como usar `describe-vpc-endpoint-service-configurations`.

AWS CLI

Para descrever as configurações do serviço de endpoint

O `describe-vpc-endpoint-service-configurations` exemplo a seguir descreve suas configurações de serviço de endpoint.

```
aws ec2 describe-vpc-endpoint-service-configurations
```

Saída:

```
{
  "ServiceConfigurations": [
    {
      "ServiceType": [
        {
          "ServiceType": "GatewayLoadBalancer"
        }
      ],
      "ServiceId": "vpce-svc-012d33a1c4321cabc",
      "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-012d33a1c4321cabc",
      "ServiceState": "Available",
      "AvailabilityZones": [
        "us-east-1d"
      ],
      "AcceptanceRequired": false,
      "ManagesVpcEndpoints": false,
    }
  ]
}
```

```

    "GatewayLoadBalancerArns": [
      "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
gwy/GWLBSservice/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": []
  }
]
}

```

Para obter mais informações, consulte [serviços de VPC endpoint](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DescribeVpcEndpointServiceConfigurations](#) na Referência de AWS CLI Comandos.

describe-vpc-endpoint-service-permissions

O código de exemplo a seguir mostra como usar `describe-vpc-endpoint-service-permissions`.

AWS CLI

Para descrever as permissões do serviço de endpoint

Este exemplo descreve as permissões para o serviço de endpoint especificado.

Comando:

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
```

Saída:

```
{
  "AllowedPrincipals": [
    {
      "PrincipalType": "Account",
      "Principal": "arn:aws:iam::123456789012:root"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeVpcEndpointServicePermissions](#) na Referência de AWS CLI Comandos.

describe-vpc-endpoint-services

O código de exemplo a seguir mostra como usar `describe-vpc-endpoint-services`.

AWS CLI

Exemplo 1: Para descrever todos os serviços de VPC endpoint

O exemplo "describe-vpc-endpoint-services" a seguir lista todos os serviços de VPC endpoint de uma AWS região.

```
aws ec2 describe-vpc-endpoint-services
```


Saída:

```
{
  "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"
]
}
```

Para obter mais informações, consulte [Exibir nomes AWS de serviços disponíveis](#) no Guia do usuário do AWS PrivateLink.

Exemplo 2: Para descrever os detalhes sobre um serviço de endpoint

O exemplo "describe-vpc-endpoint-services" a seguir lista os detalhes do serviço de endpoint da interface Amazon S3

```
aws ec2 describe-vpc-endpoint-services \
  --filter "Name=service-type,Values=Interface" Name=service-
  name,Values=com.amazonaws.us-east-1.s3
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Exibir nomes AWS de serviços disponíveis](#) no Guia do usuário do AWS PrivateLink.

- Para API obter detalhes, consulte [DescribeVpcEndpointServices](#) na Referência de AWS CLI Comandos.

describe-vpc-endpoints

O código de exemplo a seguir mostra como usar `describe-vpc-endpoints`.

AWS CLI

Para descrever seus VPC endpoints

O `describe-vpc-endpoints` exemplo a seguir exibe detalhes de todos os seus VPC endpoints.

```
aws ec2 describe-vpc-endpoints
```

Saída:

```
{
  "VpcEndpoints": [
    {
      "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}",
      "VpcId": "vpc-aabb1122",
      "NetworkInterfaceIds": [],
      "SubnetIds": [],
      "PrivateDnsEnabled": true,
      "State": "available",
      "ServiceName": "com.amazonaws.us-east-1.dynamodb",
      "RouteTableIds": [
        "rtb-3d560345"
      ],
      "Groups": [],
      "VpcEndpointId": "vpce-032a826a",
      "VpcEndpointType": "Gateway",
      "CreationTimestamp": "2017-09-05T20:41:28Z",
      "DnsEntries": [],
      "OwnerId": "123456789012"
    },
    {
```

```

    "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*\n\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource\n\": \"*\">\n    ]\n  }",
    "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"
}
]
```

Para obter mais informações, consulte [VPCendpoints](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DescribeVpcEndpoints](#) na Referência de AWS CLI Comandos.

describe-vpc-peering-connections

O código de exemplo a seguir mostra como usar `describe-vpc-peering-connections`.

AWS CLI

Para descrever suas conexões de VPC peering

Este exemplo descreve todas as suas conexões de VPC peering.

Comando:

```
aws ec2 describe-vpc-peering-connections
```

Saída:

```
{
```

```
"VpcPeeringConnections": [
  {
    "Status": {
      "Message": "Active",
      "Code": "active"
    },
    "Tags": [
      {
        "Value": "Peering-1",
        "Key": "Name"
      }
    ],
    "AccepterVpcInfo": {
      "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",
        "AccepterVpcInfo": {
            "OwnerId": "444455556666",
            "VpcId": "vpc-33cc44dd"
        }
    ]
}

```

Para descrever conexões de VPC emparelhamento específicas

Este exemplo descreve todas as suas conexões de VPC peering que estão no estado de aceitação pendente.

Comando:

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

Este exemplo descreve todas as suas conexões de VPC peering que têm a tag Owner=Finance.

Comando:

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

Este exemplo descreve todas as conexões de VPC peering que você solicitou para o especificadoVPC, vpc-1a2b3c4d.

Comando:

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- Para API obter detalhes, consulte [DescribeVpcPeeringConnections](#) na Referência de AWS CLI Comandos.

describe-vpcs

O código de exemplo a seguir mostra como usar describe-vpcs.

AWS CLI

Exemplo 1: Para descrever todas as suas VPCs

O `describe-vpcs` exemplo a seguir recupera detalhes sobre seu VPCs.

```
aws ec2 describe-vpcs
```

Saída:

```
{
  "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"
      }
    ]
  }
]
}

```

Exemplo 2: Para descrever um determinado VPC

O `describe-vpcs` exemplo a seguir recupera detalhes do especificado VPC.

```

aws ec2 describe-vpcs \
  --vpc-ids vpc-06e4ab6c6cEXAMPLE

```

Saída:

```

{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeVpcs](#) na Referência de AWS CLI Comandos.

describe-vpn-connections

O código de exemplo a seguir mostra como usar `describe-vpn-connections`.

AWS CLI

Exemplo 1: Para descrever suas VPN conexões

O `describe-vpn-connections` exemplo a seguir descreve todas as suas conexões de site VPN a site.

```
aws ec2 describe-vpn-connections
```

Saída:

```

{
  "VpnConnections": [
    {
      "CustomerGatewayConfiguration": "...configuration information...",
      "CustomerGatewayId": "cgw-01234567abcde1234",
      "Category": "VPN",
      "State": "available",
      "Type": "ipsec.1",
      "VpnConnectionId": "vpn-1122334455aabbccd",
      "TransitGatewayId": "tgw-00112233445566aab",
    }
  ]
}

```

```

    "Options": {
      "EnableAcceleration": false,
      "StaticRoutesOnly": true,
      "LocalIpv4NetworkCidr": "0.0.0.0/0",
      "RemoteIpv4NetworkCidr": "0.0.0.0/0",
      "TunnelInsideIpVersion": "ipv4"
    },
    "Routes": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "CanadaVPN"
      }
    ],
    "VgwTelemetry": [
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2020-07-29T10:35:11.000Z",
        "OutsideIpAddress": "203.0.113.3",
        "Status": "DOWN",
        "StatusMessage": ""
      },
      {
        "AcceptedRouteCount": 0,
        "LastStatusChange": "2020-09-02T09:09:33.000Z",
        "OutsideIpAddress": "203.0.113.5",
        "Status": "UP",
        "StatusMessage": ""
      }
    ]
  ]
}

```

Para obter mais informações, consulte [Como VPN funciona AWS de site a site no Guia do usuário de site](#) a AWS site VPN.

Exemplo 2: Para descrever suas VPN conexões disponíveis

O `describe-vpn-connections` exemplo a seguir descreve suas VPN conexões de site a site com um estado de `available`

```
aws ec2 describe-vpn-connections \
```

```
--filters "Name=state,Values=available"
```

Para obter mais informações, consulte [Como VPN funciona AWS de site a site no Guia do usuário de site](#) a AWS site VPN.

- Para API obter detalhes, consulte [DescribeVpnConnections](#) na Referência de AWS CLI Comandos.

describe-vpn-gateways

O código de exemplo a seguir mostra como usar `describe-vpn-gateways`.

AWS CLI

Para descrever seus gateways privados virtuais

Este exemplo descreve seus gateways privados virtuais.

Comando:

```
aws ec2 describe-vpn-gateways
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [DescribeVpnGateways](#) na Referência de AWS CLI Comandos.

detach-classic-link-vpc

O código de exemplo a seguir mostra como usar `detach-classic-link-vpc`.

AWS CLI

Para desvincular (separar) uma instância EC2 -Classic de um VPC

Este exemplo desvincula a instância `i-0598c7d356eba48d7` do `vpc-88888888`. VPC

Comando:

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-id vpc-88888888
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [DetachClassicLinkVpc](#) na Referência de AWS CLI Comandos.

detach-internet-gateway

O código de exemplo a seguir mostra como usar `detach-internet-gateway`.

AWS CLI

Para separar um gateway de internet do seu VPC

O `detach-internet-gateway` exemplo a seguir separa o gateway de internet especificado do específicoVPC.

```
aws ec2 detach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Gateways de Internet](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DetachInternetGateway](#) na Referência de AWS CLI Comandos.

detach-network-interface

O código de exemplo a seguir mostra como usar `detach-network-interface`.

AWS CLI

Para separar uma interface de rede da sua instância

Este exemplo separa a interface de rede especificada da instância especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- Para API obter detalhes, consulte [DetachNetworkInterface](#) na Referência de AWS CLI Comandos.

detach-verified-access-trust-provider

O código de exemplo a seguir mostra como usar `detach-verified-access-trust-provider`.

AWS CLI

Para separar um provedor confiável de uma instância

O `detach-verified-access-trust-provider` exemplo a seguir separa o provedor confiável de acesso verificado especificado da instância de acesso verificado especificada.

```
aws ec2 detach-verified-access-trust-provider \  
--verified-access-instance-id vai-0ce000c0b7643abea \  
--verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Instâncias de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [DetachVerifiedAccessTrustProvider](#) na Referência de AWS CLI Comandos.

detach-volume

O código de exemplo a seguir mostra como usar detach-volume.

AWS CLI

Para separar um volume de uma instância

Este exemplo de comando separa o volume (vol-049df61146c4d7901) da instância à qual ele está conectado.

Comando:

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

Saída:

```
{
  "AttachTime": "2014-02-27T19:23:06.000Z",
  "InstanceId": "i-1234567890abcdef0",
  "VolumeId": "vol-049df61146c4d7901",
  "State": "detaching",
  "Device": "/dev/sdb"
}
```

- Para API obter detalhes, consulte [DetachVolume](#) na Referência de AWS CLI Comandos.

detach-vpn-gateway

O código de exemplo a seguir mostra como usar `detach-vpn-gateway`.

AWS CLI

Para separar um gateway privado virtual do seu VPC

Este exemplo separa o gateway privado virtual especificado do especificado VPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- Para API obter detalhes, consulte [DetachVpnGateway](#) na Referência de AWS CLI Comandos.

disable-address-transfer

O código de exemplo a seguir mostra como usar `disable-address-transfer`.

AWS CLI

Para desativar uma transferência de endereço IP elástico

O `disable-address-transfer` exemplo a seguir desativa a transferência de endereço IP elástico para o endereço IP elástico especificado.

```
aws ec2 disable-address-transfer \  
  --allocation-id eipalloc-09ad461b0d03f6aaf
```

Saída:

```
{  
  "AddressTransfer": {  
    "PublicIp": "100.21.184.216",  
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
    "AddressTransferStatus": "disabled"  
  }  
}
```

Para obter mais informações, consulte [Transferir endereços IP elásticos](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [DisableAddressTransfer](#) na Referência de AWS CLI Comandos.

disable-aws-network-performance-metric-subscription

O código de exemplo a seguir mostra como usar `disable-aws-network-performance-metric-subscription`.

AWS CLI

Para desativar uma assinatura métrica

O `disable-aws-network-performance-metric-subscription` exemplo a seguir desativa o monitoramento da latência agregada da rede entre as regiões de origem e destino especificadas.

```
aws ec2 disable-aws-network-performance-metric-subscription \  
  --source us-east-1 \  
  --destination eu-west-1 \  
  --metric aggregate-latency \  
  --statistic p50
```

Saída:

```
{
  "Output": true
}
```

Para obter mais informações, consulte [Gerenciar assinaturas no Guia](#) do usuário de desempenho da infraestrutura.

- Para API obter detalhes, consulte [DisableAwsNetworkPerformanceMetricSubscription](#) na Referência de AWS CLI Comandos.

disable-ebs-encryption-by-default

O código de exemplo a seguir mostra como usar `disable-ebs-encryption-by-default`.

AWS CLI

Para desativar a EBS criptografia por padrão

O `disable-ebs-encryption-by-default` exemplo a seguir desativa a EBS criptografia por padrão para sua AWS conta na região atual.

```
aws ec2 disable-ebs-encryption-by-default
```

Saída:

```
{
  "EbsEncryptionByDefault": false
}
```

- Para API obter detalhes, consulte [DisableEbsEncryptionByDefault](#) na Referência de AWS CLI Comandos.

disable-fast-launch

O código de exemplo a seguir mostra como usar `disable-fast-launch`.

AWS CLI

Para interromper o lançamento rápido de uma imagem

O `disable-fast-launch` exemplo a seguir interrompe a inicialização rápida no especificado AMI e limpa os snapshots pré-provisionados existentes.

```
aws ec2 disable-fast-launch \  
  --image-id ami-01234567890abcdef
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre como configurar um Windows AMI para um lançamento mais rápido, consulte [Configurar seu AMI para um lançamento mais rápido](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [DisableFastLaunch](#) na Referência de AWS CLI Comandos.

disable-fast-snapshot-restores

O código de exemplo a seguir mostra como usar `disable-fast-snapshot-restores`.

AWS CLI

Para desativar a restauração rápida de instantâneos

O `disable-fast-snapshot-restores` exemplo a seguir desativa a restauração rápida do instantâneo para o instantâneo especificado na zona de disponibilidade especificada.

```
aws ec2 disable-fast-snapshot-restores \  
  --availability-zones us-east-2a \  
  --source-snapshot-ids snap-1234567890abcdef0
```

Saída:

```
{  
  "Successful": [  
    {  
      "SnapshotId": "snap-1234567890abcdef0"  
      "AvailabilityZone": "us-east-2a",  
      "State": "disabling",  
      "StateTransitionReason": "Client.UserInitiated",  
      "OwnerId": "123456789012",  
      "EnablingTime": "2020-01-25T23:57:49.602Z"  
    }  
  ],  
  "Unsuccessful": []  
}
```

- Para API obter detalhes, consulte [DisableFastSnapshotRestores](#) na Referência de AWS CLI Comandos.

disable-image-block-public-access

O código de exemplo a seguir mostra como usar `disable-image-block-public-access`.

AWS CLI

Para desativar o bloqueio do acesso público AMIs na região especificada

O `disable-image-block-public-access` exemplo a seguir desativa o bloqueio do acesso público AMIs no nível da conta na região especificada.

```
aws ec2 disable-image-block-public-access \  
  --region us-east-1
```

Saída:

```
{  
  "ImageBlockPublicAccessState": "unblocked"
```

```
}
```

Para obter mais informações, consulte [Bloquear o acesso público ao seu AMIs](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DisableImageBlockPublicAccess](#) na Referência de AWS CLI Comandos.

disable-image-deprecation

O código de exemplo a seguir mostra como usar `disable-image-deprecation`.

AWS CLI

Para cancelar a suspensão de uso de um AMI

O `disable-image-deprecation` exemplo a seguir cancela a suspensão de uso de um AMI, o que remove o `DeprecationTime` campo da saída. `describe-images` Você deve ser o AMI proprietário para realizar esse procedimento.

```
aws ec2 disable-image-deprecation \
  --image-id ami-1234567890abcdef0
```

Saída:

```
{
  "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",
  "Return": "true"
}
```

Para obter mais informações, consulte [Descontinuar um AMI <https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html>](https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/ami-deprecate.html) no Guia do usuário da Amazon. EC2

- Para API obter detalhes, consulte [DisableImageDeprecation](#) na Referência de AWS CLI Comandos.

disable-image

O código de exemplo a seguir mostra como usar `disable-image`.

AWS CLI

Para desativar um AMI

O `disable-image` exemplo a seguir desativa o especificado AMI.

```
aws ec2 disable-image \  
  --image-id ami-1234567890abcdef0
```

Saída:

```
{  
  "Return": "true"  
}
```

Para obter mais informações, consulte [Desativar um AMI](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DisableImage](#) na Referência de AWS CLI Comandos.

disable-ipam-organization-admin-account

O código de exemplo a seguir mostra como usar `disable-ipam-organization-admin-account`.

AWS CLI

Para desativar o administrador delegado IPAM

Em determinados cenários, você se integrará IPAM ao AWS Organizations. Quando você faz isso, a conta de gerenciamento do AWS Organizations delega uma conta de membro do AWS Organizations como IPAM administrador.

Neste exemplo, você é a conta de gerenciamento da AWS Organizations que delegou a conta de IPAM administrador e deseja impedir que essa conta seja a IPAM administradora.

Você pode usar qualquer AWS região para `--region` fazer essa solicitação. Você não precisa usar a região na qual delegou originalmente o administrador, onde IPAM ele foi criado ou uma região IPAM operacional. Se você desativar a conta de administrador delegado, poderá reativá-la a qualquer momento ou delegar uma nova conta como administrador. IPAM

O `disable-ipam-organization-admin-account` exemplo a seguir desativa o IPAM administrador delegado em sua AWS conta.

```
aws ec2 disable-ipam-organization-admin-account \  
  --delegated-admin-account-id 320805250157 \  
  --region ap-south-1
```

Saída:

```
{  
  "Success": true  
}
```

Para obter mais informações, consulte [Integrar IPAM com contas em uma AWS organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DisableIpamOrganizationAdminAccount](#) na Referência de AWS CLI Comandos.

disable-serial-console-access

O código de exemplo a seguir mostra como usar `disable-serial-console-access`.

AWS CLI

Para desativar o acesso ao console EC2 serial da sua conta

O `disable-serial-console-access` exemplo a seguir desativa o acesso da conta ao console serial.

```
aws ec2 disable-serial-console-access
```

Saída:

```
{  
  "SerialConsoleAccessEnabled": false  
}
```

Para obter mais informações, consulte [EC2Serial Console](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DisableSerialConsoleAccess](#) na Referência de AWS CLI Comandos.

disable-snapshot-block-public-access

O código de exemplo a seguir mostra como usar `disable-snapshot-block-public-access`.

AWS CLI

Para desativar o bloqueio do acesso público para instantâneos

O `disable-snapshot-block-public-access` exemplo a seguir desativa o bloqueio do acesso público para instantâneos para permitir o compartilhamento público de seus instantâneos.

```
aws ec2 disable-snapshot-block-public-access
```

Saída:

```
{
  "State": "unblocked"
}
```

Para obter mais informações, consulte [Bloquear o acesso público para snapshots](#) no Guia do EBS usuário da Amazon.

- Para API obter detalhes, consulte [DisableSnapshotBlockPublicAccess](#) na Referência de AWS CLI Comandos.

disable-transit-gateway-route-table-propagation

O código de exemplo a seguir mostra como usar `disable-transit-gateway-route-table-propagation`.

AWS CLI

Para desativar um anexo de gateway de trânsito para propagar rotas para a tabela de rotas de propagação especificada

O `disable-transit-gateway-route-table-propagation` exemplo a seguir desativa o anexo especificado para propagar rotas para a tabela de rotas de propagação especificada.

```
aws ec2 disable-transit-gateway-route-table-propagation \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Saída:

```
{
  "Propagation": {
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
    "ResourceId": "vpc-4d7de228",
    "ResourceType": "vpc",
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
    "State": "disabled"
  }
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DisableTransitGatewayRouteTablePropagation](#) na Referência de AWS CLI Comandos.

disable-vgw-route-propagation

O código de exemplo a seguir mostra como usar `disable-vgw-route-propagation`.

AWS CLI

Para desativar a propagação da rota

Este exemplo impede que o gateway privado virtual especificado propague rotas estáticas para a tabela de rotas especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- Para API obter detalhes, consulte [DisableVgwRoutePropagation](#) na Referência de AWS CLI Comandos.

disable-vpc-classic-link-dns-support

O código de exemplo a seguir mostra como usar `disable-vpc-classic-link-dns-support`.

AWS CLI

Para desativar o ClassicLink DNS suporte para um VPC

Este exemplo desativa o ClassicLink DNS suporte para `vpc-88888888`.

Comando:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [DisableVpcClassicLinkDnsSupport](#) na Referência de AWS CLI Comandos.

disable-vpc-classic-link

O código de exemplo a seguir mostra como usar `disable-vpc-classic-link`.

AWS CLI

Para desativar ClassicLink para um VPC

Este exemplo desativa ClassicLink para `vpc-88888888`.

Comando:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [DisableVpcClassicLink](#) na Referência de AWS CLI Comandos.

disassociate-address

O código de exemplo a seguir mostra como usar `disassociate-address`.

AWS CLI

Para desassociar um endereço IP elástico em EC2 -Classic

Este exemplo dissocia um endereço IP elástico de uma instância em EC2 -Classic. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

Para desassociar um endereço IP elástico em EC2 - VPC

Este exemplo dissocia um endereço IP elástico de uma instância em um VPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- Para API obter detalhes, consulte [DisassociateAddress](#) na Referência de AWS CLI Comandos.

disassociate-client-vpn-target-network

O código de exemplo a seguir mostra como usar `disassociate-client-vpn-target-network`.

AWS CLI

Para desassociar uma rede de um endpoint do cliente VPN

O `disassociate-client-vpn-target-network` exemplo a seguir desassocia a rede de destino associada à ID de `cvpn-assoc-12312312312312312` associação do endpoint do cliente VPN especificado.

```
aws ec2 disassociate-client-vpn-target-network \  
--client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  

```

```
--association-id cvpn-assoc-12312312312312312
```

Saída:

```
{
  "AssociationId": "cvpn-assoc-12312312312312312",
  "Status": {
    "Code": "disassociating"
  }
}
```

Para obter mais informações, consulte [Target Networks](#) no Guia AWS do VPN administrador do cliente.

- Para API obter detalhes, consulte [DisassociateClientVpnTargetNetwork](#) na Referência de AWS CLI Comandos.

disassociate-iam-instance-profile

O código de exemplo a seguir mostra como usar `disassociate-iam-instance-profile`.

AWS CLI

Para desassociar um perfil de IAM instância

Este exemplo desassocia um perfil de IAM instância com o ID da associação. `iip-assoc-05020b59952902f5f`

Comando:

```
aws ec2 disassociate-iam-instance-profile --association-id iip-  
assoc-05020b59952902f5f
```

Saída:

```
{
  "IamInstanceProfileAssociation": {
    "InstanceId": "i-123456789abcde123",
    "State": "disassociating",
    "AssociationId": "iip-assoc-05020b59952902f5f",
    "IamInstanceProfile": {
      "Id": "AIPAI5IVIHMFY2DKV5Y",

```

```

    "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
  }
}
}

```

- Para API obter detalhes, consulte [Disassocie uma instância de perfil de instância](#) na Referência de AWS CLI Comandos.

disassociate-instance-event-window

O código de exemplo a seguir mostra como usar `disassociate-instance-event-window`.

AWS CLI

Exemplo 1: Para desassociar uma ou mais instâncias de uma janela de eventos

O `disassociate-instance-event-window` exemplo a seguir dissocia uma ou mais instâncias de uma janela de eventos. Especifique o `instance-event-window-id` parâmetro para especificar a janela do evento. Para dissociar instâncias, especifique o `association-target` parâmetro e, para os valores dos parâmetros, especifique uma ou mais instâncias IDs.

```

aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"

```

Saída:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 2: Para desassociar tags de instância de uma janela de eventos

O `disassociate-instance-event-window` exemplo a seguir dissocia as tags de instância de uma janela de eventos. Especifique o `instance-event-window-id` parâmetro para especificar a janela do evento. Para desassociar tags de instância, especifique o `association-target` para os valores de parâmetro, especifique uma ou mais tags.

```
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}]"
```

Saída:

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 3: Para desassociar um host dedicado de uma janela de eventos

O `disassociate-instance-event-window` exemplo a seguir desassocia um host dedicado de uma janela de eventos. Especifique o `instance-event-window-id` parâmetro para especificar a janela do evento. Para desassociar um host dedicado, especifique o `association-target` parâmetro e, para os valores dos parâmetros, especifique um ou mais host IDs dedicado.

```
aws ec2 disassociate-instance-event-window \  
--region us-east-1 \  
--instance-event-window-id iew-0abcdef1234567890 \  
--association-target DedicatedHostIds=h-029fa35a02b99801d
```

Saída:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [DisassociateInstanceEventWindow](#) na Referência de AWS CLI Comandos.

disassociate-ipam-resource-discovery

O código de exemplo a seguir mostra como usar `disassociate-ipam-resource-discovery`.

AWS CLI

Para dissociar uma descoberta de recursos de um IPAM

Neste exemplo, você tem uma conta de administrador IPAM delegado e deseja desassociar uma descoberta de IPAM recursos da sua. IPAM Você executou o comando `describe` e percebeu que o `"ResourceDiscoveryStatus": "not-found"` e deseja desassociá-lo do seu IPAM para abrir espaço para outras associações.

O `disassociate-ipam-resource-discovery` exemplo a seguir desassocia uma descoberta IPAM de recursos em sua AWS conta.


```
aws ec2 disassociate-ipam-resource-discovery \
  --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82
\
  --region us-east-1
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [DisassociateIpamResourceDiscovery](#) na Referência de AWS CLI Comandos.

disassociate-nat-gateway-address

O código de exemplo a seguir mostra como usar `disassociate-nat-gateway-address`.

AWS CLI

Para desassociar um endereço IP elástico de um gateway público NAT

O `disassociate-nat-gateway-address` exemplo a seguir desassocia o endereço IP elástico especificado do gateway público NAT especificado.

```
aws ec2 disassociate-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --association-ids eipassoc-0f96bdca17EXAMPLE
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [DisassociateNatGatewayAddress](#) na Referência de AWS CLI Comandos.

disassociate-route-table

O código de exemplo a seguir mostra como usar `disassociate-route-table`.

AWS CLI

Para desassociar uma tabela de rotas

Este exemplo dissocia a tabela de rotas especificada da sub-rede especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- Para API obter detalhes, consulte [DisassociateRouteTable](#) na Referência de AWS CLI Comandos.

disassociate-subnet-cidr-block

O código de exemplo a seguir mostra como usar `disassociate-subnet-cidr-block`.

AWS CLI

Para desassociar um IPv6 CIDR bloco de uma sub-rede

Este exemplo dissocia um IPv6 CIDR bloco de uma sub-rede usando o ID de associação do bloco. CIDR

Comando:

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

Saída:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
    "AssociationId": "subnet-cidr-assoc-3aa54053",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  }
}
```

- Para API obter detalhes, consulte [DisassociateSubnetCidrBlock](#) na Referência de AWS CLI Comandos.

disassociate-transit-gateway-multicast-domain

O código de exemplo a seguir mostra como usar `disassociate-transit-gateway-multicast-domain`.

AWS CLI

Para desassociar sub-redes de um domínio multicast

O `disassociate-transit-gateway-multicast-domain` exemplo a seguir desassocia uma sub-rede do domínio multicast especificado.

```
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
```

Saída:

```
{  
  "Associations": [  
    {  
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",  
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",  
      "ResourceId": "vpc-7EXAMPLE",  
      "ResourceType": "vpc",  
      "Subnets": [  
        {  
          "SubnetId": "subnet-000de86e3bEXAMPLE",  
          "State": "disassociating"  
        }  
      ]  
    }  
  ]  
}
```

Para obter mais informações, consulte [Trabalhando com multicast](#) no Guia de Gateways de Trânsito '.

- Para API obter detalhes, consulte [DisassociateTransitGatewayMulticastDomain](#) na Referência de AWS CLI Comandos.

disassociate-transit-gateway-route-table

O código de exemplo a seguir mostra como usar `disassociate-transit-gateway-route-table`.

AWS CLI

Para desassociar uma tabela de rotas do Transit Gateway de um anexo de recurso

O `disassociate-transit-gateway-route-table` exemplo a seguir desassocia o anexo especificado da tabela de rotas do gateway de trânsito.

```
aws ec2 disassociate-transit-gateway-route-table \  
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE
```

Saída:

```
{  
  "Association": {  
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",  
    "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",  
    "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",  
    "ResourceType": "direct-connect-gateway",  
    "State": "disassociating"  
  }  
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [DisassociateTransitGatewayRouteTable](#) na Referência de AWS CLI Comandos.

disassociate-vpc-cidr-block

O código de exemplo a seguir mostra como usar `disassociate-vpc-cidr-block`.

AWS CLI

Para desassociar um IPv6 CIDR bloco de um VPC

Este exemplo dissocia um IPv6 CIDR bloco de um VPC usando o ID de associação do CIDR bloco.

Comando:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

Saída:

```
{
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
    "AssociationId": "vpc-cidr-assoc-eca54085",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  },
  "VpcId": "vpc-a034d6c4"
}
```

Para desassociar um IPv4 CIDR bloco de um VPC

Este exemplo dissocia um IPv4 CIDR bloco de um VPC

Comando:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

Saída:

```
{
  "CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-0287ac6b",
    "CidrBlock": "172.18.0.0/16",
    "CidrBlockState": {
      "State": "disassociating"
    }
  },
  "VpcId": "vpc-27621243"
}
```

- Para API obter detalhes, consulte [DisassociateVpcCidrBlock](#) na Referência de AWS CLI Comandos.

enable-address-transfer

O código de exemplo a seguir mostra como usar `enable-address-transfer`.

AWS CLI

Para habilitar uma transferência de endereço IP elástico

O `enable-address-transfer` exemplo a seguir permite a transferência do endereço IP elástico do endereço IP elástico especificado para a conta especificada.

```
aws ec2 enable-address-transfer \
  --allocation-id eipalloc-09ad461b0d03f6aaf \
  --transfer-account-id 123456789012
```

Saída:

```
{
  "AddressTransfer": {
    "PublicIp": "100.21.184.216",
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",
    "TransferAccountId": "123456789012",
    "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",
    "AddressTransferStatus": "pending"
  }
}
```

Para obter mais informações, consulte [Transferir endereços IP elásticos](#) no Guia VPC do usuário da Amazon.

- Para API obter detalhes, consulte [EnableAddressTransfer](#) na Referência de AWS CLI Comandos.

enable-aws-network-performance-metric-subscription

O código de exemplo a seguir mostra como usar `enable-aws-network-performance-metric-subscription`.

AWS CLI

Para habilitar uma assinatura métrica

O `enable-aws-network-performance-metric-subscription` exemplo a seguir permite o monitoramento da latência agregada da rede entre as regiões de origem e destino especificadas.

```
aws ec2 enable-aws-network-performance-metric-subscription \  
  --source us-east-1 \  
  --destination eu-west-1 \  
  --metric aggregate-latency \  
  --statistic p50
```

Saída:

```
{  
  "Output": true  
}
```

Para obter mais informações, consulte [Gerenciar assinaturas no Guia](#) do usuário de desempenho da infraestrutura.

- Para API obter detalhes, consulte [EnableAwsNetworkPerformanceMetricSubscription](#) na Referência de AWS CLI Comandos.

enable-ebs-encryption-by-default

O código de exemplo a seguir mostra como usar `enable-ebs-encryption-by-default`.

AWS CLI

Para habilitar a EBS criptografia por padrão

O `enable-ebs-encryption-by-default` exemplo a seguir ativa a EBS criptografia por padrão para sua AWS conta na região atual.

```
aws ec2 enable-ebs-encryption-by-default
```

Saída:

```
{  
  "EbsEncryptionByDefault": true  
}
```


- Para API obter detalhes, consulte [EnableEbsEncryptionByDefault](#) na Referência de AWS CLI Comandos.

enable-fast-launch

O código de exemplo a seguir mostra como usar `enable-fast-launch`.

AWS CLI

Para iniciar o lançamento rápido de uma imagem

O `enable-fast-launch` exemplo a seguir inicia a inicialização rápida no especificado AMI e define o número máximo de instâncias paralelas a serem executadas como 6. O tipo de recurso a ser usado para pré-provisionar o AMI está definido como `snapshot`, que também é o valor padrão.

```
aws ec2 enable-fast-launch \  
  --image-id ami-01234567890abcdef \  
  --max-parallel-launches 6 \  
  --resource-type snapshot
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre como configurar um Windows AMI para um lançamento mais rápido, consulte [Configurar seu AMI para um lançamento mais rápido](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [EnableFastLaunch](#) na Referência de AWS CLI Comandos.

enable-fast-snapshot-restores

O código de exemplo a seguir mostra como usar `enable-fast-snapshot-restores`.

AWS CLI

Para ativar a restauração rápida de instantâneos

O `enable-fast-snapshot-restores` exemplo a seguir permite a restauração rápida de instantâneos para o instantâneo especificado nas zonas de disponibilidade especificadas.

```
aws ec2 enable-fast-snapshot-restores \
  --availability-zones us-east-2a us-east-2b \
  --source-snapshot-ids snap-1234567890abcdef0
```

Saída:

```
{
  "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": []
}
```

- Para API obter detalhes, consulte [EnableFastSnapshotRestores](#) na Referência de AWS CLI Comandos.

enable-image-block-public-access

O código de exemplo a seguir mostra como usar `enable-image-block-public-access`.

AWS CLI

Para habilitar o bloqueio do acesso público AMIs na região especificada

O `enable-image-block-public-access` exemplo a seguir permite bloquear o acesso público AMIs no nível da conta na região especificada.

```
aws ec2 enable-image-block-public-access \  
  --region us-east-1 \  
  --image-block-public-access-state block-new-sharing
```

Saída:

```
{  
  "ImageBlockPublicAccessState": "block-new-sharing"  
}
```

Para obter mais informações, consulte [Bloquear o acesso público ao seu AMIs](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [EnableImageBlockPublicAccess](#) na Referência de AWS CLI Comandos.

enable-image-deprecation

O código de exemplo a seguir mostra como usar `enable-image-deprecation`.

AWS CLI

Exemplo 1: Para descontinuar um AMI

O `enable-image-deprecation` exemplo a seguir desaprova um AMI em uma data e hora específicas. Se você especificar um valor para segundos, a Amazon EC2 arredondará os segundos para o minuto mais próximo. Você deve ser o AMI proprietário para realizar esse procedimento.

```
aws ec2 enable-image-deprecation \  
  --image-id ami-1234567890abcdef0 \  
  --deprecation-date 2017-01-01T00:00:00 \  
  --seconds 30
```

```
--deprecate-at "2022-10-15T13:17:12.000Z"
```

Saída:

```
{
  "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",
  "Return": "true"
}
```

Para obter mais informações, consulte [Descontinuar um AMI](https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/deprecate-ami) <<https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/deprecate-ami>> no Guia do usuário da Amazon. EC2

- Para API obter detalhes, consulte [EnableImageDeprecation](#) na Referência de AWS CLI Comandos.

enable-image

O código de exemplo a seguir mostra como usar `enable-image`.

AWS CLI

Para habilitar um AMI

O `enable-image` exemplo a seguir ativa o especificado AMI.

```
aws ec2 enable-image \
  --image-id ami-1234567890abcdef0
```

Saída:

```
{
  "Return": "true"
}
```

Para obter mais informações, consulte [Desativar um AMI](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [EnableImage](#) na Referência de AWS CLI Comandos.

enable-ipam-organization-admin-account

O código de exemplo a seguir mostra como usar `enable-ipam-organization-admin-account`.

AWS CLI

Para integrar-se com AWS Organizations e delegar uma conta de membro como conta IPAM

O `enable-ipam-organization-admin-account` exemplo a seguir se integra IPAM ao AWS Organizations e delega uma conta de membro como a IPAM conta.

```
aws ec2 enable-ipam-organization-admin-account \  
--delegated-admin-account-id 320805250157
```

Saída:

```
{  
  "Success": true  
}
```

Para obter mais informações, consulte [Integrate IPAM with AWS Organizations](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [EnableIpamOrganizationAdminAccount](#) na Referência de AWS CLI Comandos.

enable-reachability-analyzer-organization-sharing

O código de exemplo a seguir mostra como usar `enable-reachability-analyzer-organization-sharing`.

AWS CLI

Para habilitar o acesso confiável ao Reachability Analyzer

O `enable-reachability-analyzer-organization-sharing` exemplo a seguir permite acesso confiável ao Reachability Analyzer.

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

Este comando não produz saída.

Para obter mais informações, consulte [Análises entre contas no Guia](#) do Usuário do Reachability Analyzer.

- Para API obter detalhes, consulte [EnableReachabilityAnalyzerOrganizationSharing](#) na Referência de AWS CLI Comandos.

enable-serial-console-access

O código de exemplo a seguir mostra como usar `enable-serial-console-access`.

AWS CLI

Para habilitar o acesso ao console serial da sua conta

O `enable-serial-console-access` exemplo a seguir permite o acesso da conta ao console serial.

```
aws ec2 enable-serial-console-access
```

Saída:

```
{
  "SerialConsoleAccessEnabled": true
}
```

Para obter mais informações, consulte [EC2Serial Console](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [EnableSerialConsoleAccess](#) na Referência de AWS CLI Comandos.

enable-snapshot-block-public-access

O código de exemplo a seguir mostra como usar `enable-snapshot-block-public-access`.

AWS CLI

Para habilitar o bloqueio do acesso público para instantâneos

O `enable-snapshot-block-public-access` exemplo a seguir bloqueia todo o compartilhamento público de seus instantâneos.

```
aws ec2 enable-snapshot-block-public-access \
  --state block-all-sharing
```

Saída:

```
{
  "State": "block-all-sharing"
}
```

Para obter mais informações, consulte [Bloquear o acesso público para snapshots](#) no Guia do EBS usuário da Amazon.

- Para API obter detalhes, consulte [EnableSnapshotBlockPublicAccess](#) na Referência de AWS CLI Comandos.

enable-transit-gateway-route-table-propagation

O código de exemplo a seguir mostra como usar `enable-transit-gateway-route-table-propagation`.

AWS CLI

Para permitir que um anexo de gateway de trânsito propague rotas para a tabela de rotas de propagação especificada

O `enable-transit-gateway-route-table-propagation` exemplo a seguir permite que o anexo especificado propague rotas para a tabela de rotas de propagação especificada.

```
aws ec2 enable-transit-gateway-route-table-propagation \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Saída:

```
{
  "Propagation": {
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
    "ResourceId": "vpc-4d7de228",
    "ResourceType": "vpc",
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
    "State": "disabled"
  }
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [EnableTransitGatewayRouteTablePropagation](#) na Referência de AWS CLI Comandos.

enable-vgw-route-propagation

O código de exemplo a seguir mostra como usar `enable-vgw-route-propagation`.

AWS CLI

Para ativar a propagação da rota

Este exemplo permite que o gateway privado virtual especificado propague rotas estáticas para a tabela de rotas especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- Para API obter detalhes, consulte [EnableVgwRoutePropagation](#) na Referência de AWS CLI Comandos.

enable-volume-io

O código de exemplo a seguir mostra como usar `enable-volume-io`.

AWS CLI

Para habilitar a E/S para um volume

Este exemplo habilita a E/S no volume `vol-1234567890abcdef0`.

Comando:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

Saída:


```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [EnableVolumelona](#) Referência de AWS CLI Comandos.

enable-vpc-classic-link-dns-support

O código de exemplo a seguir mostra como usar `enable-vpc-classic-link-dns-support`.

AWS CLI

Para habilitar o ClassicLink DNS suporte para um VPC

Este exemplo habilita ClassicLink DNS o suporte paravpc-88888888.

Comando:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [EnableVpcClassicLinkDnsSupport](#)na Referência de AWS CLI Comandos.

enable-vpc-classic-link

O código de exemplo a seguir mostra como usar `enable-vpc-classic-link`.

AWS CLI

Para habilitar um VPC para ClassicLink

Este exemplo habilita o vpc-88888888 para. ClassicLink

Comando:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [EnableVpcClassicLink](#) na Referência de AWS CLI Comandos.

export-client-vpn-client-certificate-revocation-list

O código de exemplo a seguir mostra como usar `export-client-vpn-client-certificate-revocation-list`.

AWS CLI

Para exportar uma lista de revogação de certificados de clientes

O `export-client-vpn-client-certificate-revocation-list` exemplo a seguir exporta a lista de revogação de certificados de cliente para o endpoint de cliente VPN especificado. Neste exemplo, a saída é retornada em formato de texto para facilitar a leitura.

```
aws ec2 export-client-vpn-client-certificate-revocation-list \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --output text
```

Saída:

```
-----BEGIN X509 CRL-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wZG8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
```

```
Ibb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END X509 CRL-----
STATUS      pending
```

Para obter mais informações, consulte [Listas de revogação de certificados de cliente](#) no Guia do VPNadministrador do AWS cliente.

- Para API obter detalhes, consulte [ExportClientVpnClientCertificateRevocationList](#) na Referência de AWS CLI Comandos.

export-client-vpn-client-configuration

O código de exemplo a seguir mostra como usar `export-client-vpn-client-configuration`.

AWS CLI

Para exportar a configuração do cliente

O `export-client-vpn-client-configuration` exemplo a seguir exporta a configuração do cliente para o VPN endpoint do cliente especificado. Neste exemplo, a saída é retornada em formato de texto para facilitar a leitura.

```
aws ec2 export-client-vpn-client-configuration \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --output text
```

Saída:

```
client
dev tun
proto udp
remote cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-south-1.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
cipher AES-256-GCM
```

```

verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBIDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIIJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----
</ca>
reneg-sec 0

```

Para obter mais informações, consulte [Client VPN Endpoints](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [ExportClientVpnClientConfiguration](#) na Referência de AWS CLI Comandos.

export-image

O código de exemplo a seguir mostra como usar `export-image`.

AWS CLI

Para exportar uma VM de um AMI

O `export-image` exemplo a seguir exporta o especificado AMI para o bucket especificado no formato especificado.

```

aws ec2 export-image \
  --image-id ami-1234567890abcdef0 \
  --disk-image-format VMDK \
  --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/

```

Saída:

```
{
  "DiskImageFormat": "vmdk",
  "ExportImageTaskId": "export-ami-1234567890abcdef0"
  "ImageId": "ami-1234567890abcdef0",
  "RoleName": "vmimport",
  "Progress": "0",
  "S3ExportLocation": {
    "S3Bucket": "my-export-bucket",
    "S3Prefix": "exports/"
  },
  "Status": "active",
  "StatusMessage": "validating"
}
```

- Para API obter detalhes, consulte [ExportImagem](#) na Referência de AWS CLI Comandos.

get-associated-ipv6-pool-cidrs

O código de exemplo a seguir mostra como usar `get-associated-ipv6-pool-cidrs`.

AWS CLI

Para obter as associações para um pool IPv6 de endereços

O `get-associated-ipv6-pool-cidrs` exemplo a seguir obtém as associações para o pool IPv6 de endereços especificado.

```
aws ec2 get-associated-ipv6-pool-cidrs \
  --pool-id ipv6pool-ec2-012345abc12345abc
```

Saída:

```
{
  "Ipv6CidrAssociations": [
    {
      "Ipv6Cidr": "2001:db8:1234:1a00::/56",
      "AssociatedResource": "vpc-111111222222333ab"
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [GetAssociatedIpv6 PoolCidrs](#) em Referência de AWS CLI comandos.

get-aws-network-performance-data

O código de exemplo a seguir mostra como usar `get-aws-network-performance-data`.

AWS CLI

Para obter dados de desempenho da rede

O `get-aws-network-performance-data` exemplo a seguir recupera dados sobre o desempenho da rede entre as regiões especificadas no período de tempo especificado.

```
aws ec2 get-aws-network-performance-data \  
  --start-time 2022-10-26T12:00:00.000Z \  
  --end-time 2022-10-26T12:30:00.000Z \  
  --data-queries Id=my-query,Source=us-east-1,Destination=eu-  
west-1,Metric=aggregate-latency,Statistic=p50,Period=five-minutes
```

Saída:

```
{  
  "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"
  }
]
}
```

Para obter mais informações, consulte [Monitorar o desempenho da rede](#) no Guia do usuário de desempenho da infraestrutura.

- Para API obter detalhes, consulte [GetAwsNetworkPerformanceData](#) na Referência de AWS CLI Comandos.

get-capacity-reservation-usage

O código de exemplo a seguir mostra como usar `get-capacity-reservation-usage`.

AWS CLI

Para ver o uso da reserva de capacidade em todas AWS as contas

O `get-capacity-reservation-usage` exemplo a seguir exibe informações de uso para a reserva de capacidade especificada.

```
aws ec2 get-capacity-reservation-usage \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Saída:

```
{
  "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
  "InstanceUsages": [
    {
      "UsedInstanceCount": 1,
      "AccountId": "123456789012"
    }
  ],
  "AvailableInstanceCount": 4,
  "TotalInstanceCount": 5,
  "State": "active",
  "InstanceType": "t2.medium"
}
```

Para obter mais informações, consulte [Visualizar o uso da reserva de capacidade compartilhada](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [GetCapacityReservationUsage](#) na Referência de AWS CLI Comandos.

get-coip-pool-usage

O código de exemplo a seguir mostra como usar `get-coip-pool-usage`.

AWS CLI

Para obter o uso do pool de endereços IP de propriedade do cliente

O `get-coip-pool-usage` exemplo a seguir obtém os detalhes de uso do pool de endereços IP de propriedade do cliente especificado.

```
aws ec2 get-coip-pool-usage \  
  --pool-id ipv4pool-coip-123a45678bEXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Endereços IP pertencentes ao cliente](#) no Guia do usuário do AWS Outposts.

- Para API obter detalhes, consulte [GetCoipPoolUsage](#) na Referência de AWS CLI Comandos.

get-console-output

O código de exemplo a seguir mostra como usar `get-console-output`.

AWS CLI

Exemplo 1: Para obter a saída do console

O `get-console-output` exemplo a seguir obtém a saída do console para a instância Linux especificada.

```
aws ec2 get-console-output \  
  --instance-id i-1234567890abcdef0
```

Saída:

```
{  
  "InstanceId": "i-1234567890abcdef0",  
  "Timestamp": "2013-07-25T21:23:53.000Z",  
  "Output": "..."  
}
```

Para obter mais informações, consulte a [saída do console de instância](#) no Amazon EC2 User Guide.

Exemplo 2: Para obter a saída mais recente do console

O `get-console-output` exemplo a seguir obtém a saída mais recente do console para a instância Linux especificada.

```
aws ec2 get-console-output \  
  --instance-id i-1234567890abcdef0 \  
  --latest \  
  --output text
```

Saída:

```
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
```

Para obter mais informações, consulte a [saída do console de instância](#) no Amazon EC2 User Guide.

- Para API obter detalhes, consulte [GetConsoleOutput](#) na Referência de AWS CLI Comandos.

get-console-screenshot

O código de exemplo a seguir mostra como usar `get-console-screenshot`.

AWS CLI

Para recuperar uma captura de tela de uma instância em execução

O `get-console-screenshot` exemplo a seguir recupera uma captura de tela da instância especificada no formato.jpg. A captura de tela é retornada como uma string codificada em Base64.

```
aws ec2 get-console-screenshot \  
  --instance-id i-1234567890abcdef0
```

Saída:

```
{  
  "ImageData": "997987/8kgj49ikjhewkwe0008084EXAMPLE",  
  "InstanceId": "i-1234567890abcdef0"  
}
```

- Para API obter detalhes, consulte [GetConsoleScreenshot](#) na Referência de AWS CLI Comandos.

get-default-credit-specification

O código de exemplo a seguir mostra como usar `get-default-credit-specification`.

AWS CLI

Para descrever a opção de crédito padrão

O `get-default-credit-specification` exemplo a seguir descreve a opção de crédito padrão para instâncias T2.

```
aws ec2 get-default-credit-specification \
  --instance-family t2
```

Saída:

```
{
  "InstanceFamilyCreditSpecification": {
    "InstanceFamily": "t2",
    "CpuCredits": "standard"
  }
}
```

- Para API obter detalhes, consulte [GetDefaultCreditSpecification](#) na Referência de AWS CLI Comandos.

get-ebs-default-kms-key-id

O código de exemplo a seguir mostra como usar `get-ebs-default-kms-key-id`.

AWS CLI

Para descrever seu padrão CMK de EBS criptografia

O `get-ebs-default-kms-key-id` exemplo a seguir descreve o padrão CMK de EBS criptografia da sua AWS conta.

```
aws ec2 get-ebs-default-kms-key-id
```

A saída mostra o padrão CMK para EBS criptografia, que é AWS gerenciado CMK com o `alias/aws/ebs`.

```
{
  "KmsKeyId": "alias/aws/ebs"
}
```

A saída a seguir mostra um personalizado CMK para EBS criptografia.

```
{
```

```
"KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- Para API obter detalhes, consulte [GetEbsDefaultKmsKeyId](#) na Referência de AWS CLI Comandos.

get-ebs-encryption-by-default

O código de exemplo a seguir mostra como usar `get-ebs-encryption-by-default`.

AWS CLI

Para descrever se a EBS criptografia por padrão está habilitada

O `get-ebs-encryption-by-default` exemplo a seguir indica se a EBS criptografia, por padrão, está habilitada para sua AWS conta na região atual.

```
aws ec2 get-ebs-encryption-by-default
```

A saída a seguir indica que a EBS criptografia, por padrão, está desativada.

```
{
  "EbsEncryptionByDefault": false
}
```

A saída a seguir indica que a EBS criptografia, por padrão, está ativada.

```
{
  "EbsEncryptionByDefault": true
}
```

- Para API obter detalhes, consulte [GetEbsEncryptionByDefault](#) na Referência de AWS CLI Comandos.

get-flow-logs-integration-template

O código de exemplo a seguir mostra como usar `get-flow-logs-integration-template`.

AWS CLI

Para criar um CloudFormation modelo para automatizar a integração dos registros de VPC fluxo com o Amazon Athena

Os `get-flow-logs-integration-template` exemplos a seguir criam um CloudFormation modelo para automatizar a integração dos registros de VPC fluxo com o 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
```

Saída:

```
{
  "Result": "https://DOC-EXAMPLE-BUCKET.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_fl-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
}
```

Para obter informações sobre o uso CloudFormation de modelos, consulte Como [trabalhar com AWS CloudFormation modelos](#) no Guia AWS CloudFormation do usuário.

Para obter informações sobre como usar o Amazon Athena e os registros de fluxo, consulte [Consultar registros de fluxo usando o Amazon Athena](#) no Guia do usuário da Amazon Virtual Private Cloud.

- Para API obter detalhes, consulte [GetFlowLogsIntegrationTemplate](#) na Referência de AWS CLI Comandos.

get-groups-for-capacity-reservation

O código de exemplo a seguir mostra como usar `get-groups-for-capacity-reservation`.

AWS CLI

Para listar os grupos de recursos com uma reserva de capacidade

O `get-groups-for-capacity-reservation` exemplo a seguir lista os grupos de recursos aos quais a reserva de capacidade especificada foi adicionada.

```
aws ec2 get-groups-for-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Saída:

```
{
  "CapacityReservationsGroup": [
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
      "OwnerId": "123456789012"
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com reservas de capacidade](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [GetGroupsForCapacityReservation](#) na Referência de AWS CLI Comandos.

get-host-reservation-purchase-preview

O código de exemplo a seguir mostra como usar `get-host-reservation-purchase-preview`.

AWS CLI

Para obter uma prévia da compra de uma reserva de anfitrião dedicado

Este exemplo fornece uma prévia dos custos de uma reserva de host dedicado especificada para o host dedicado especificado em sua conta.

Comando:

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --  
host-id-set h-013abcd2a00cbd123
```

Saída:

```
{  
  "TotalHourlyPrice": "1.499",  
  "Purchase": [  
    {  
      "HourlyPrice": "1.499",  
      "InstanceFamily": "m4",  
      "PaymentOption": "NoUpfront",  
      "HostIdSet": [  
        "h-013abcd2a00cbd123"  
      ],  
      "UpfrontPrice": "0.000",  
      "Duration": 31536000  
    }  
  ],  
  "TotalUpfrontPrice": "0.000"  
}
```

- Para API obter detalhes, consulte [GetHostReservationPurchasePreview](#) na Referência de AWS CLI Comandos.

get-image-block-public-access-state

O código de exemplo a seguir mostra como usar `get-image-block-public-access-state`.

AWS CLI

Para obter o estado de bloqueio de acesso público AMLs na região especificada

O `get-image-block-public-access-state` exemplo a seguir obtém o estado de bloqueio de acesso público AMIs no nível da conta na região especificada.

```
aws ec2 get-image-block-public-access-state \  
  --region us-east-1
```

Saída:

```
{  
  "ImageBlockPublicAccessState": "block-new-sharing"  
}
```

Para obter mais informações, consulte [Bloquear o acesso público ao seu AMIs](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [GetImageBlockPublicAccessState](#) na Referência de AWS CLI Comandos.

get-instance-types-from-instance-requirements

O código de exemplo a seguir mostra como usar `get-instance-types-from-instance-requirements`.

AWS CLI

Para visualizar os tipos de instância que correspondem aos atributos especificados

O `get-instance-types-from-instance-requirements` exemplo a seguir primeiro gera uma lista de todos os atributos possíveis que podem ser especificados usando o `--generate-cli-skeleton` parâmetro e salva a lista em um JSON arquivo. Em seguida, o JSON arquivo é usado para personalizar os atributos para os quais visualizar os tipos de instância correspondentes.

Para gerar todos os atributos possíveis e salvar a saída diretamente em um JSON arquivo, use o comando a seguir.

```
aws ec2 get-instance-types-from-instance-requirements \  
  --region us-east-1 \  
  --generate-cli-skeleton input > attributes.json
```

Saída:

```
{
  "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": ""
}

```

Configure o arquivo JSON. Você deve fornecer valores para `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` e `MemoryMiB`. Você pode omitir os outros atributos. Quando omitidos, os valores padrão são usados. Para obter uma descrição de cada atributo e seus valores padrão, consulte `get-instance-types-from -instance-requirements < latest/reference/ec2/ -instance-requirements.html >`. <https://docs.aws.amazon.com/cli/get-instance-types-from>

Visualize os tipos de instância que têm os atributos especificados em `attributes.json`. Especifique o nome e o caminho para seu JSON arquivo usando o `--cli-input-json` parâmetro. Na solicitação a seguir, a saída é formatada como uma tabela.

```
aws ec2 get-instance-types-from-instance-requirements \
```

```
--cli-input-json file://attributes.json \  
--output table
```

Conteúdo do arquivo `attributes.json`:

```
{  
  
  "ArchitectureTypes": [  
    "x86_64"  
  ],  
  "VirtualizationTypes": [  
    "hvm"  
  ],  
  "InstanceRequirements": {  
    "VCpuCount": {  
      "Min": 4,  
      "Max": 6  
    },  
    "MemoryMiB": {  
      "Min": 2048  
    },  
    "InstanceGenerations": [  
      "current"  
    ]  
  }  
}
```

Saída:

```
-----  
|GetInstanceTypesFromInstanceRequirements|  
+-----+  
||           InstanceTypes           ||  
|+-----+|  
||           InstanceType           ||  
|+-----+|  
|| c4.xlarge                          ||  
|| c5.xlarge                          ||  
|| c5a.xlarge                         ||  
|| c5ad.xlarge                        ||  
|| c5d.xlarge                         ||  
|| c5n.xlarge                         ||  
|| d2.xlarge                          ||  
|+-----+|  
-----
```

...

Para obter mais informações sobre a seleção de tipo de instância baseada em atributos, consulte [Como funciona a seleção de tipo de instância baseada em atributos no Guia do usuário](#) da Amazon. EC2

- Para API obter detalhes, consulte [GetInstanceTypesFromInstanceRequirements](#) na Referência de AWS CLI Comandos.

get-instance-uefi-data

O código de exemplo a seguir mostra como usar `get-instance-uefi-data`.

AWS CLI

Para recuperar UEFI dados de uma instância

O `get-instance-uefi-data` exemplo a seguir recupera UEFI dados de uma instância. Se a saída estiver vazia, a instância não conterá UEFI dados.

```
aws ec2 get-instance-uefi-data \
  --instance-id i-0123456789example
```

Saída:

```
{
  "InstanceId": "i-0123456789example",
  "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.
  <snipped>
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/A0Dshho="
}
```

Para obter mais informações, consulte [UEFI Secure Boot](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [GetInstanceUefiData](#) na Referência de AWS CLI Comandos.

get-ipam-address-history

O código de exemplo a seguir mostra como usar `get-ipam-address-history`.

AWS CLI

Para obter a história de um CIDR

O `get-ipam-address-history` exemplo a seguir obtém o histórico de um 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
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [Visualizar o histórico de endereços IP](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [GetIppamAddressHistory](#) na Referência de AWS CLI Comandos.

get-ippam-discovered-accounts

O código de exemplo a seguir mostra como usar `get-ippam-discovered-accounts`.

AWS CLI

Para ver as contas descobertas por um IPAM

Nesse cenário, você é um administrador IPAM delegado que deseja visualizar as AWS contas que possuem recursos que ele IPAM está descobrindo.

Essa `--discovery-region` é a região IPAM operacional na qual você deseja visualizar os status da conta monitorada. Por exemplo, se você tiver três regiões IPAM operacionais, talvez

queira fazer essa solicitação três vezes para ver os timestamps específicos da descoberta em cada uma dessas regiões específicas.

O `get-ipam-discovered-accounts` exemplo a seguir lista as AWS contas que possuem recursos que a IPAM está descobrindo.

```
aws ec2 get-ipam-discovered-accounts \
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
  --discovery-region us-east-1
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Integrar IPAM com contas fora da sua organização](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [GetIpamDiscoveredAccounts](#) na Referência de AWS CLI Comandos.

get-ipam-discovered-public-addresses

O código de exemplo a seguir mostra como usar `get-ipam-discovered-public-addresses`.

AWS CLI

Para visualizar endereços IP públicos descobertos

Neste exemplo, você é um administrador IPAM delegado e deseja visualizar os endereços IP dos recursos descobertos pelo IPAM. Você pode obter o ID de descoberta do recurso com [describe-ipam-resource-discoveries](#).

O `get-ipam-discovered-public-addresses` exemplo a seguir mostra os endereços IP públicos descobertos para uma descoberta de recursos.

```
aws ec2 get-ipam-discovered-public-addresses \  
  --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \  
  --address-region us-east-1 \  
  --region us-east-1
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Exibir insights de IP público](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [GetIpamDiscoveredPublicAddresses](#) na Referência de AWS CLI Comandos.

get-ipam-discovered-resource-cidrs

O código de exemplo a seguir mostra como usar `get-ipam-discovered-resource-cidrs`.

AWS CLI

Para ver o endereço IP CIDRs descoberto por um IPAM

Neste exemplo, você é um administrador IPAM delegado que deseja ver detalhes relacionados ao endereço CIDRs IP dos recursos que IPAM está descobrindo.

Para concluir essa solicitação:

A descoberta de recursos que você escolher deve estar associada à IPAM. `--resource-region` É a AWS região em que o recurso foi criado.

O `get-ipam-discovered-resource-cidrs` exemplo a seguir lista os endereços IP dos recursos que ele IPAM está descobrindo.

```

aws ec2 get-ipam-discovered-resource-cidrs \
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \

```

```
--resource-region us-east-1
```

Saída:

```
{
  {
    "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"
      }
    ]
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Monitorar o CIDR uso por recurso](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [GetIpamDiscoveredResourceCidrs](#) na Referência de AWS CLI Comandos.

get-ipam-pool-allocations

O código de exemplo a seguir mostra como usar `get-ipam-pool-allocations`.

AWS CLI

Para obter a CIDRs alocação de um pool IPAM

O `get-ipam-pool-allocations` exemplo a seguir obtém a CIDRs alocação de um IPAM pool.

(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
```

Saída:

```
{  
  "IpamPoolAllocations": [  
    {  
      "Cidr": "10.0.0.0/16",  
      "IpamPoolAllocationId": "ipam-pool-alloc-0e6186d73999e47389266a5d6991e6220",
```

```

        "ResourceType": "custom",
        "ResourceOwner": "123456789012"
    }
]
}

```

- Para API obter detalhes, consulte [GetIpamPoolAllocations](#) na Referência de AWS CLI Comandos.

get-ipam-pool-cidrs

O código de exemplo a seguir mostra como usar `get-ipam-pool-cidrs`.

AWS CLI

Para obter o CIDRs aprovisionamento em um pool IPAM

O `get-ipam-pool-cidrs` exemplo a seguir fornece o CIDRs provisionamento para um 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.*

```

Saída:

```

{
  "IpamPoolCidr": {
    "Cidr": "10.0.0.0/24",
    "State": "provisioned"
  }
}

```

- Para API obter detalhes, consulte [GetIpamPoolCidrs](#) na Referência de AWS CLI Comandos.

get-ipam-resource-cidrs

O código de exemplo a seguir mostra como usar `get-ipam-resource-cidrs`.

AWS CLI

Para obter o CIDRs alocado para um recurso

O `get-ipam-resource-cidrs` exemplo a seguir obtém o CIDRs alocado para um recurso.

(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
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [Monitorar o CIDR uso por recurso](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [GetIpamResourceCidrs](#) na Referência de AWS CLI Comandos.

get-launch-template-data

O código de exemplo a seguir mostra como usar `get-launch-template-data`.

AWS CLI

Para obter dados de instância para um modelo de lançamento

Este exemplo obtém dados sobre a instância especificada e usa a `--query` opção para retornar o conteúdo `LaunchTemplateData`. É possível usar o resultado como base para criar um novo modelo de execução ou uma versão de modelo de execução.

Comando:

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
'LaunchTemplateData'
```

Saída:

```
{
  "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"
        }
    ]
}

```

- Para API obter detalhes, consulte [GetLaunchTemplateData](#) na Referência de AWS CLI Comandos.

get-managed-prefix-list-associations

O código de exemplo a seguir mostra como usar `get-managed-prefix-list-associations`.

AWS CLI

Para obter associações de listas de prefixos

O `get-managed-prefix-list-associations` exemplo a seguir obtém os recursos associados à lista de prefixos especificada.

```
aws ec2 get-managed-prefix-list-associations \  
  --prefix-list-id p1-0123456abcabcabc1
```

Saída:

```
{  
  "PrefixListAssociations": [  
    {  
      "ResourceId": "sg-0abc123456abc12345",  
      "ResourceOwner": "123456789012"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [GetManagedPrefixListAssociations](#) na Referência de AWS CLI Comandos.

get-managed-prefix-list-entries

O código de exemplo a seguir mostra como usar `get-managed-prefix-list-entries`.

AWS CLI

Para obter as entradas de uma lista de prefixos

O seguinte `get-managed-prefix-list-entries` obtém as entradas para a lista de prefixos especificada.

```
aws ec2 get-managed-prefix-list-entries \  
  --prefix-list-id p1-0123456abcabcabc1
```

Saída:

```
{
  "Entries": [
    {
      "Cidr": "10.0.0.0/16",
      "Description": "vpc-a"
    },
    {
      "Cidr": "10.2.0.0/16",
      "Description": "vpc-b"
    }
  ]
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [GetManagedPrefixListEntries](#) na Referência de AWS CLI Comandos.

get-network-insights-access-scope-analysis-findings

O código de exemplo a seguir mostra como usar `get-network-insights-access-scope-analysis-findings`.

AWS CLI

Para obter as conclusões da análise do escopo de acesso do Network Insights

O `get-network-insights-access-scope-analysis-findings` exemplo a seguir obtém os resultados da análise de escopo selecionada em sua AWS conta.

```
aws ec2 get-network-insights-access-scope-analysis-findings \
  --region us-east-1 \
  --network-insights-access-scope-analysis-id nisa \
  --nis-123456789111
```

Saída:

```
{
  "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
  "AnalysisFindings": [
    {
```

```
"NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",
"NetworkInsightsAccessScopeId": "nis-123456789111",
"FindingComponents": [
  {
    "SequenceNumber": 1,
    "Component": {
      "Id": "eni-02e3d42d5cceca67d",
      "Arn": "arn:aws:ec2:us-east-1:936459623503:network-
interface/eni-02e3d32d9cceca17d"
    },
    "OutboundHeader": {
      "DestinationAddresses": [
        "0.0.0.0/5",
        "11.0.0.0/8",
        "12.0.0.0/6",
        "128.0.0.0/3",
        "16.0.0.0/4",
        "160.0.0.0/5",
        "168.0.0.0/6",
        "172.0.0.0/12"
        "8.0.0.0/7"
      ],
      "DestinationPortRanges": [
        {
          "From": 0,
          "To": 65535
        }
      ],
      "Protocol": "6",
      "SourceAddresses": [
        "10.0.2.253/32"
      ],
      "SourcePortRanges": [
        {
          "From": 0,
          "To": 65535
        }
      ]
    }, [etc]
  }
]
```

```
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI Network Access Analyzer](#).

- Para API obter detalhes, consulte [GetNetworkInsightsAccessScopeAnalysisFindings](#) na Referência de AWS CLI Comandos.

get-network-insights-access-scope-content

O código de exemplo a seguir mostra como usar `get-network-insights-access-scope-content`.

AWS CLI

Para obter o Network Insights, acesse o conteúdo do escopo

O `get-network-insights-access-scope-content` exemplo a seguir obtém o conteúdo da ID de análise de escopo selecionada em sua AWS conta.

```
aws ec2 get-network-insights-access-scope-content \
  --region us-east-1 \
  --network-insights-access-scope-id nis-123456789222
```

Saída:

```
{
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-123456789222",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::NetworkInterface"
            ]
          }
        },
        "Destination": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
}
```


Comando:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0 --priv-launch-key C:\Keys\MyKeyPair.pem
```

Saída:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-30T23:18:05.000Z",
  "PasswordData": "&ViJ652e*u"
}
```

- Para API obter detalhes, consulte [GetPasswordData](#) na Referência de AWS CLI Comandos.

get-reserved-instances-exchange-quote

O código de exemplo a seguir mostra como usar `get-reserved-instances-exchange-quote`.

AWS CLI

Para obter uma cotação para trocar uma instância reservada conversível

Este exemplo obtém as informações de troca para as instâncias reservadas conversíveis especificadas.

Comando:

```
aws ec2 get-reserved-instances-exchange-quote --reserved-instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

Saída:

```
{
  "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"
  }
}

```

- Para API obter detalhes, consulte [GetReservedInstancesExchangeQuota](#) na Referência de AWS CLI Comandos.

get-security-groups-for-vpc

O código de exemplo a seguir mostra como usar `get-security-groups-for-vpc`.

AWS CLI

Para visualizar grupos de segurança que podem ser associados a interfaces de rede em um determinado VPC.

O `get-security-groups-for-vpc` exemplo a seguir mostra os grupos de segurança que podem ser associados às interfaces de rede noVPC.

```
aws ec2 get-security-groups-for-vpc \  
  --vpc-id vpc-6c31a611 \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [GetSecurityGroupsForVpc](#) na Referência de AWS CLI Comandos.

get-serial-console-access-status

O código de exemplo a seguir mostra como usar `get-serial-console-access-status`.

AWS CLI

Para ver o status do acesso da conta ao console serial

O `get-serial-console-access-status` exemplo a seguir determina se o acesso ao console serial está habilitado para sua conta.


```
aws ec2 get-serial-console-access-status
```

Saída:

```
{
  "SerialConsoleAccessEnabled": true
}
```

Para obter mais informações, consulte [EC2Serial Console](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [GetSerialConsoleAccessStatus](#) na Referência de AWS CLI Comandos.

get-snapshot-block-public-access-state

O código de exemplo a seguir mostra como usar `get-snapshot-block-public-access-state`.

AWS CLI

Para obter o estado atual do bloqueio de acesso público para instantâneos

O `get-snapshot-block-public-access-state` exemplo a seguir mostra o estado atual do bloqueio de acesso público para snapshots.

```
aws ec2 get-snapshot-block-public-access-state
```

Saída:

```
{
  "State": "block-all-sharing"
}
```

Para obter mais informações, consulte [Bloquear o acesso público para snapshots](#) no Guia do EBS usuário da Amazon.

- Para API obter detalhes, consulte [GetSnapshotBlockPublicAccessState](#) na Referência de AWS CLI Comandos.

get-spot-placement-scores

O código de exemplo a seguir mostra como usar `get-spot-placement-scores`.

AWS CLI

Para calcular a pontuação de colocação do Spot para requisitos especificados

O `get-spot-placement-scores` exemplo a seguir primeiro gera uma lista de todos os parâmetros possíveis que podem ser especificados para a configuração da pontuação de posicionamento do Spot usando o `--generate-cli-skeleton` parâmetro e salva a lista em um JSON arquivo. Em seguida, o JSON arquivo é usado para configurar os requisitos a serem usados para calcular a pontuação de posicionamento do Spot.

Para gerar todos os parâmetros possíveis que podem ser especificados para a configuração da pontuação de posicionamento do Spot e salvar a saída diretamente em um JSON arquivo.

```
aws ec2 get-spot-placement-scores \
  --region us-east-1 \
  --generate-cli-skeleton input > attributes.json
```

Saída:

```
{
  "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": ""
}

```

Configure o arquivo JSON. Você deve fornecer um valor para `TargetCapacity`: Para obter uma descrição de cada parâmetro e seus valores padrão, consulte [Calcular a pontuação de posicionamento do Spot \(AWS CLI\)](https://docs.aws.amazon.com/AWS EC2/latest//.html#UserGuide) <<https://docs.aws.amazon.com/AWS EC2/latest//.html#UserGuide>>. `spot-placement-score calculate-sps-cli`

Calcule a pontuação de colocação do Spot para os requisitos especificados em `attributes.json`. Especifique o nome e o caminho para seu JSON arquivo usando o `--cli-input-json` parâmetro.

```

aws ec2 get-spot-placement-scores \
  --region us-east-1 \
  --cli-input-json file://attributes.json

```

Saída se `SingleAvailabilityZone` for definida como `false` ou omitida (se omitida, o padrão é). `false` Uma lista pontuada de regiões é retornada.

```

"Recommendation": [
  {
    "Region": "us-east-1",
    "Score": 7
  },
  {
    "Region": "us-west-1",
    "Score": 5
  },
]

```

```
...
```

Saída se `SingleAvailabilityZone` estiver definida como `true`. Uma lista `SingleAvailability` pontuada de zonas é retornada.

```
"Recommendation": [
  {
    "Region": "us-east-1",
    "AvailabilityZoneId": "use1-az1"
    "Score": 8
  },
  {
    "Region": "us-east-1",
    "AvailabilityZoneId": "usw2-az3"
    "Score": 6
  },
  ...
```

Para obter mais informações sobre o cálculo de uma pontuação de colocação spot e, por exemplo, configurações, consulte [Calcular uma pontuação de colocação spot no Guia EC2](#) do usuário da Amazon.

- Para API obter detalhes, consulte [GetSpotPlacementScores](#) na Referência de AWS CLI Comandos.

get-subnet-cidr-reservations

O código de exemplo a seguir mostra como usar `get-subnet-cidr-reservations`.

AWS CLI

Para obter informações sobre uma reserva de sub-rede CIDR

O `get-subnet-cidr-reservations` exemplo a seguir exibe informações sobre a CIDR reserva de sub-rede especificada.

```
aws ec2 get-subnet-cidr-reservations \
  --subnet-id subnet-03c51e2e6cEXAMPLE
```

Saída:

```
{
```

```

    "SubnetIpv4CidrReservations": [
      {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
      }
    ],
    "SubnetIpv6CidrReservations": []
  }

```

Para obter mais informações, consulte [CIDRReservas de sub-rede](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [GetSubnetCidrReservations](#) na Referência de AWS CLI Comandos.

get-transit-gateway-attachment-propagations

O código de exemplo a seguir mostra como usar `get-transit-gateway-attachment-propagations`.

AWS CLI

Para listar as tabelas de rotas para as quais o anexo de recurso especificado propaga rotas

O `get-transit-gateway-attachment-propagations` exemplo a seguir lista a tabela de rotas para a qual o anexo de recurso especificado propaga rotas.

```

aws ec2 get-transit-gateway-attachment-propagations \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE

```

Saída:

```

{
  "TransitGatewayAttachmentPropagations": [
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
      "State": "enabled"
    }
  ]
}

```

```
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [GetTransitGatewayAttachmentPropagations](#) na Referência de AWS CLI Comandos.

get-transit-gateway-multicast-domain-associations

O código de exemplo a seguir mostra como usar `get-transit-gateway-multicast-domain-associations`.

AWS CLI

Para ver as informações sobre as associações de domínio multicast do Transit Gateway

O `get-transit-gateway-multicast-domain-associations` exemplo a seguir retorna as associações para o domínio multicast especificado.

```
aws ec2 get-transit-gateway-multicast-domain-associations \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Saída:

```
{  
  "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"
      }
    }
  ]
}

```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [GetTransitGatewayMulticastDomainAssociations](#) na Referência de AWS CLI Comandos.

get-transit-gateway-prefix-list-references

O código de exemplo a seguir mostra como usar `get-transit-gateway-prefix-list-references`.

AWS CLI

Para obter referências da lista de prefixos em uma tabela de rotas do Transit Gateway

O `get-transit-gateway-prefix-list-references` exemplo a seguir obtém as referências da lista de prefixos para a tabela de rotas do Transit Gateway especificada e filtra pelo ID de uma lista de prefixos específica.

```
aws ec2 get-transit-gateway-prefix-list-references \  
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
  --filters Name=prefix-list-id,Values=pl-111112222222333
```

Saída:

```
{  
  "TransitGatewayPrefixListReferences": [  
    {  
      "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",  
      "PrefixListId": "pl-111112222222333",  
      "PrefixListOwnerId": "123456789012",  
      "State": "available",  
      "Blackhole": false,  
      "TransitGatewayAttachment": {  
        "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",  
        "ResourceType": "vpc",  
        "ResourceId": "vpc-112233445566aabbcc"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte as [referências da lista de prefixos](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [GetTransitGatewayPrefixListReferences](#) na Referência de AWS CLI Comandos.

get-transit-gateway-route-table-associations

O código de exemplo a seguir mostra como usar `get-transit-gateway-route-table-associations`.

AWS CLI

Para obter informações sobre as associações para a tabela de rotas do gateway de trânsito especificada

O `get-transit-gateway-route-table-associations` exemplo a seguir exibe informações sobre as associações da tabela de rotas do gateway de trânsito especificada.

```
aws ec2 get-transit-gateway-route-table-associations \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

Saída:

```
{
  "Associations": [
    {
      "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
      "ResourceId": "vpc-4d7de228",
      "ResourceType": "vpc",
      "State": "associating"
    }
  ]
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [GetTransitGatewayRouteTableAssociations](#) na Referência de AWS CLI Comandos.

get-transit-gateway-route-table-propagations

O código de exemplo a seguir mostra como usar `get-transit-gateway-route-table-propagations`.

AWS CLI

Para exibir informações sobre as propagações da tabela de rotas para a tabela de rotas do gateway de trânsito especificada

O `get-transit-gateway-route-table-propagations` exemplo a seguir retorna as propagações da tabela de rotas para a tabela de rotas especificada.

```
aws ec2 get-transit-gateway-route-table-propagations \
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [GetTransitGatewayRouteTablePropagations](#) na Referência de AWS CLI Comandos.

get-verified-access-endpoint-policy

O código de exemplo a seguir mostra como usar `get-verified-access-endpoint-policy`.

AWS CLI

Para obter a política de acesso verificado de um endpoint

O `get-verified-access-endpoint-policy` exemplo a seguir obtém a política de acesso verificado do endpoint especificado.

```
aws ec2 get-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2
```

Saída:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Para obter mais informações, consulte [Políticas de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [GetVerifiedAccessEndpointPolicy](#) na Referência de AWS CLI Comandos.

get-verified-access-group-policy

O código de exemplo a seguir mostra como usar `get-verified-access-group-policy`.

AWS CLI

Para obter a política de acesso verificado de um grupo

O `get-verified-access-group-policy` exemplo a seguir obtém a política de acesso verificado do grupo especificado.

```
aws ec2 get-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235
```

Saída:

```
{
  "PolicyEnabled": true,
```

```
"PolicyDocument": "permit(principal,action,resource)\nwhen\n  {\n    context.identity.groups.contains(\"finance\") &&\n    context.identity.email_verified == true\n  };"
```

Para obter mais informações, consulte [Grupos de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [GetVerifiedAccessGroupPolicy](#) na Referência de AWS CLI Comandos.

get-vpn-connection-device-sample-configuration

O código de exemplo a seguir mostra como usar `get-vpn-connection-device-sample-configuration`.

AWS CLI

Para baixar um arquivo de configuração de amostra

O `get-vpn-connection-device-sample-configuration` exemplo a seguir baixa o arquivo de configuração de amostra especificado. Para listar os dispositivos de gateway com um arquivo de configuração de amostra, chame o `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
```

Saída:

```
{
  "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-
file"
}
```

Para obter mais informações, consulte [Baixar o arquivo de configuração](#) no Guia do usuário AWS de site a site VPN.

- Para API obter detalhes, consulte [GetVpnConnectionDeviceSampleConfiguration](#) na Referência de AWS CLI Comandos.

get-vpn-connection-device-types

O código de exemplo a seguir mostra como usar `get-vpn-connection-device-types`.

AWS CLI

Para listar dispositivos de gateway com um arquivo de configuração de amostra

O `get-vpn-connection-device-types` exemplo a seguir lista os dispositivos de gateway da Palo Alto Networks que têm exemplos de arquivos de configuração.

```
aws ec2 get-vpn-connection-device-types \
  --query "VpnConnectionDeviceTypes[?Vendor=='Palo Alto Networks']"
```

Saída:

```
[
  {
    "VpnConnectionDeviceTypeId": "754a6372",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 4.1.2+"
  },
  {
    "VpnConnectionDeviceTypeId": "9612cbed",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 4.1.2+ (GUI)"
  },
  {
    "VpnConnectionDeviceTypeId": "5fb390ba",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 7.0+"
  }
]
```

Para obter mais informações, consulte [Baixar o arquivo de configuração](#) no Guia do usuário AWS de site a site VPN.

- Para API obter detalhes, consulte [GetVpnConnectionDeviceTypes](#) na Referência de AWS CLI Comandos.


```
--disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.ova}"
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [ImportImage](#) na Referência de AWS CLI Comandos.

import-key-pair

O código de exemplo a seguir mostra como usar `import-key-pair`.

AWS CLI

Para importar uma chave pública

Primeiro, gere um key pair com a ferramenta de sua escolha. Por exemplo, use este comando `ssh-keygen`:

Comando:

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

Saída:

```
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.
...
```

Este exemplo de comando importa a chave pública especificada.

Comando:

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

Saída:

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- Para API obter detalhes, consulte [ImportKeyPair](#) na Referência de AWS CLI Comandos.

import-snapshot

O código de exemplo a seguir mostra como usar `import-snapshot`.

AWS CLI

Para importar um instantâneo

O `import-snapshot` exemplo a seguir importa o disco especificado como um instantâneo.

```
aws ec2 import-snapshot \
  --description "My server VMDK" \
  --disk-container Format=VMDK,UserBucket={S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.vmdk}
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [ImportSnapshot](#) na Referência de AWS CLI Comandos.

list-images-in-recycle-bin

O código de exemplo a seguir mostra como usar `list-images-in-recycle-bin`.

AWS CLI

Para listar as imagens na Lixeira

O `list-images-in-recycle-bin` exemplo a seguir lista todas as imagens que estão atualmente retidas na Lixeira.

```
aws ec2 list-images-in-recycle-bin
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Recuperar AMIs da lixeira](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [ListImagesInRecycleBin](#) na Referência de AWS CLI Comandos.

list-snapshots-in-recycle-bin

O código de exemplo a seguir mostra como usar `list-snapshots-in-recycle-bin`.

AWS CLI

Para visualizar instantâneos na Lixeira

O `list-snapshots-in-recycle-bin` exemplo a seguir lista informações sobre instantâneos na Lixeira, incluindo a ID do instantâneo, uma descrição do instantâneo, o ID do volume do qual o instantâneo foi criado, a data e a hora em que o instantâneo foi excluído e entrou na Lixeira e a data e a hora em que o período de retenção expira.

```
aws ec2 list-snapshots-in-recycle-bin \  
--snapshot-id snap-01234567890abcdef
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações sobre a Lixeira para a AmazonEBS, consulte [Recuperar instantâneos da Lixeira no](#) Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [ListSnapshotsInRecycleBin](#) na Referência de AWS CLI Comandos.

lock-snapshot

O código de exemplo a seguir mostra como usar lock-snapshot.

AWS CLI

Exemplo 1: Para bloquear um instantâneo no modo de governança

O lock-snapshot exemplo a seguir bloqueia o instantâneo especificado no modo de governança.

```
aws ec2 lock-snapshot \  
  --snapshot-id snap-0b5e733b4a8df6e0d \  
  --lock-mode governance \  
  --lock-duration 365
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Snapshot lock](#) no Guia do EBS usuário da Amazon.

Exemplo 2: Para bloquear um instantâneo no modo de conformidade

O lock-snapshot exemplo a seguir bloqueia o instantâneo especificado no modo de conformidade.

```
aws ec2 lock-snapshot \  
  --snapshot-id snap-0163a8524c5b9901f \  
  --lock-mode compliance \  
  --cool-off-period 24 \  
  --lock-duration 365
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Snapshot lock](#) no Guia do EBS usuário da Amazon.

- Para API obter detalhes, consulte [LockSnapshot](#) na Referência de AWS CLI Comandos.

modify-address-attribute

O código de exemplo a seguir mostra como usar `modify-address-attribute`.

AWS CLI

Para modificar o atributo do nome de domínio associado a um endereço IP elástico

Os `modify-address-attribute` exemplos a seguir modificam o atributo de nome de domínio de um endereço IP elástico.

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
```

Saída:

```
{
  "Addresses": [
```

```

    {
      "PublicIp": "192.0.2.0",
      "AllocationId": "eipalloc-abcdef01234567890",
      "PtrRecord": "example.net."
      "PtrRecordUpdate": {
        "Value": "example.com.",
        "Status": "PENDING"
      }
    }
  ]
}

```

Para monitorar a alteração pendente e visualizar os atributos modificados de um endereço IP elástico, consulte [describe-addresses-attribute](#) na Referência de AWS CLI Comandos.

- Para API obter detalhes, consulte [ModifyAddressAttribute](#) na Referência de AWS CLI Comandos.

modify-availability-zone-group

O código de exemplo a seguir mostra como usar `modify-availability-zone-group`.

AWS CLI

Para habilitar um grupo de zonas

O `modify-availability-zone-group` exemplo a seguir ativa o grupo de zonas especificado.

```

aws ec2 modify-availability-zone-group \
  --group-name us-west-2-lax-1 \
  --opt-in-status opted-in

```

Saída:

```

{
  "Return": true
}

```

Para obter mais informações, consulte [Regiões e zonas](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyAvailabilityZoneGroup](#) na Referência de AWS CLI Comandos.

modify-capacity-reservation-fleet

O código de exemplo a seguir mostra como usar `modify-capacity-reservation-fleet`.

AWS CLI

Exemplo 1: Para modificar a capacidade total alvo de uma frota de reserva de capacidade

O `modify-capacity-reservation-fleet` exemplo a seguir modifica a capacidade total alvo da frota de reserva de capacidade especificada. Ao modificar a capacidade total de destino de uma frota de reserva de capacidade, a frota cria automaticamente novas reservas de capacidade ou modifica ou cancela as reservas de capacidade existentes na frota para atender à nova capacidade total de destino. Não é possível tentar fazer modificações adicionais em uma frota enquanto ela estiver no estado `modifying`.

```
aws ec2 modify-capacity-reservation-fleet \  
  --capacity-reservation-fleet-id crf-01234567890abcdef \  
  --total-target-capacity 160
```

Saída:

```
{  
  "Return": true  
}
```

Exemplo 2: Para modificar a data final de uma frota de reserva de capacidade

O `modify-capacity-reservation-fleet` exemplo a seguir modifica a data final da frota de reserva de capacidade especificada. Quando você modifica a data de término da frota, as datas de término de todas as reservas de capacidade individuais são atualizadas de maneira adequada. Não é possível tentar fazer modificações adicionais em uma frota enquanto ela estiver no estado `modifying`.

```
aws ec2 modify-capacity-reservation-fleet \  
  --capacity-reservation-fleet-id crf-01234567890abcdef \  
  --end-date 2022-07-04T23:59:59.000Z
```

Saída:

```
{
```

```
"Return": true
}
```

Para obter mais informações sobre frotas de reserva de capacidade, consulte [Frotas de reserva de capacidade](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [ModifyCapacityReservationFleet](#) na Referência de AWS CLI Comandos.

modify-capacity-reservation

O código de exemplo a seguir mostra como usar `modify-capacity-reservation`.

AWS CLI

Exemplo 1: Para alterar o número de instâncias reservadas por uma reserva de capacidade existente

O `modify-capacity-reservation` exemplo a seguir altera o número de instâncias para as quais a reserva de capacidade reserva capacidade.

```
aws ec2 modify-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE \
  --instance-count 5
```

Saída:

```
{
  "Return": true
}
```

Exemplo 2: Para alterar a data e a hora de término de uma reserva de capacidade existente

O `modify-capacity-reservation` exemplo a seguir modifica uma reserva de capacidade existente para terminar na data e hora especificadas.

```
aws ec2 modify-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE \
  --end-date-type Limited \
  --end-date 2019-08-31T23:59:59Z
```


Para obter mais informações, consulte [Modificar uma reserva de capacidade](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyCapacityReservation](#) na Referência de AWS CLI Comandos.

modify-client-vpn-endpoint

O código de exemplo a seguir mostra como usar `modify-client-vpn-endpoint`.

AWS CLI

Para modificar um VPN endpoint do cliente

O `modify-client-vpn-endpoint` exemplo a seguir ativa o registro de conexão do cliente para o VPN endpoint do cliente especificado.

```
aws ec2 modify-client-vpn-endpoint \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

Saída:

```
{
  "Return": true
}
```

Para obter mais informações, consulte [Client VPN Endpoints](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [ModifyClientVpnEndpoint](#) na Referência de AWS CLI Comandos.

modify-default-credit-specification

O código de exemplo a seguir mostra como usar `modify-default-credit-specification`.

AWS CLI

Para modificar a opção de crédito padrão

O `modify-default-credit-specification` exemplo a seguir modifica a opção de crédito padrão para instâncias T2.

```
aws ec2 modify-default-credit-specification \  
  --instance-family t2 \  
  --cpu-credits unlimited
```

Saída:

```
{  
  "InstanceFamilyCreditSpecification": {  
    "InstanceFamily": "t2",  
    "CpuCredits": "unlimited"  
  }  
}
```

- Para API obter detalhes, consulte [ModifyDefaultCreditSpecification](#) na Referência de AWS CLI Comandos.

modify-ebs-default-kms-key-id

O código de exemplo a seguir mostra como usar `modify-ebs-default-kms-key-id`.

AWS CLI

Para definir seu padrão CMK para EBS criptografia

O `modify-ebs-default-kms-key-id` exemplo a seguir define o especificado CMK como padrão CMK para EBS criptografia para sua AWS conta na região atual.

```
aws ec2 modify-ebs-default-kms-key-id \  
  --kms-key-id alias/my-cmk
```

Saída:

```
{  
  "KmsKeyId": "arn:aws:kms:us-  
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"  
}
```

- Para API obter detalhes, consulte [ModifyEbsDefaultKmsKeyId](#) na Referência de AWS CLI Comandos.

modify-fleet

O código de exemplo a seguir mostra como usar `modify-fleet`.

AWS CLI

Para escalar uma EC2 frota

O `modify-fleet` exemplo a seguir modifica a capacidade alvo da EC2 frota especificada. Se o valor especificado for maior que a capacidade atual, a EC2 frota iniciará instâncias adicionais. Se o valor especificado for menor que a capacidade atual, a EC2 frota cancela todas as solicitações abertas e, se a política de rescisão for `terminate`, a EC2 frota encerrará todas as instâncias que excederem a nova capacidade alvo.

```
aws ec2 modify-fleet \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --target-capacity-specification TotalTargetCapacity=5
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Gerenciando uma EC2 frota](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyFleet](#) na Referência de AWS CLI Comandos.

modify-fpga-image-attribute

O código de exemplo a seguir mostra como usar `modify-fpga-image-attribute`.

AWS CLI

Para modificar os atributos de uma FPGA imagem da Amazon

Este exemplo adiciona permissões de carregamento para o ID da conta 123456789012 para o especificadoAFI.

Comando:

```
aws ec2 modify-fpga-image-attribute --attribute LoadPermission --fpga-image-id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

Saída:

```
{
  "FpgaImageAttribute": {
    "FpgaImageId": "afi-0d123e123bfc85abc",
    "LoadPermissions": [
      {
        "UserId": "123456789012"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [ModifyFpgaImageAttribute](#) na Referência de AWS CLI Comandos.

modify-hosts

O código de exemplo a seguir mostra como usar `modify-hosts`.

AWS CLI

Exemplo 1: Para habilitar o posicionamento automático para um host dedicado

O `modify-hosts` exemplo a seguir permite o posicionamento automático de um host dedicado para que ele aceite qualquer execução de instância não segmentada que corresponda à configuração do tipo de instância.

```
aws ec2 modify-hosts \
  --host-id h-06c2f189b4EXAMPLE \
  --auto-placement on
```

Saída:

```
{
  "Successful": [
    "h-06c2f189b4EXAMPLE"
  ],
  "Unsuccessful": []
}
```

Exemplo 2: Para habilitar a recuperação do host para um host dedicado

O `modify-hosts` exemplo a seguir permite a recuperação do host dedicado especificado.

```
aws ec2 modify-hosts \
  --host-id h-06c2f189b4EXAMPLE \
  --host-recovery on
```

Saída:

```
{
  "Successful": [
    "h-06c2f189b4EXAMPLE"
  ],
  "Unsuccessful": []
}
```

Para obter mais informações, consulte [Modificar a colocação automática de hosts dedicados](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyHosts](#) na Referência de AWS CLI Comandos.

modify-id-format

O código de exemplo a seguir mostra como usar `modify-id-format`.

AWS CLI

Para habilitar o formato de ID mais longo para um recurso

O `modify-id-format` exemplo a seguir ativa o formato de ID mais longo para o tipo de `instance` recurso.

```
aws ec2 modify-id-format \
```

```
--resource instance \  
--use-long-ids
```

Para desativar o formato de ID mais longo para um recurso

O `modify-id-format` exemplo a seguir desativa o formato de ID mais longo para o tipo `instance` de recurso.

```
aws ec2 modify-id-format \  
--resource instance \  
--no-use-long-ids
```

O `modify-id-format` exemplo a seguir ativa o formato de ID mais longo para todos os tipos de recursos compatíveis que estão dentro do período de aceitação.

```
aws ec2 modify-id-format \  
--resource all-current \  
--use-long-ids
```

- Para API obter detalhes, consulte [ModifyIdFormat](#) na Referência de AWS CLI Comandos.

modify-identity-id-format

O código de exemplo a seguir mostra como usar `modify-identity-id-format`.

AWS CLI

Para permitir que uma IAM função seja usada IDs por mais tempo para um recurso

O `modify-identity-id-format` exemplo a seguir permite que a IAM função `EC2Role` em sua AWS conta use o formato de ID longo para o tipo de `instance` recurso.

```
aws ec2 modify-identity-id-format \  
--principal-arn arn:aws:iam::123456789012:role/EC2Role \  
--resource instance \  
--use-long-ids
```

Para permitir que um IAM usuário use um recurso IDs por mais tempo

O `modify-identity-id-format` exemplo a seguir permite que o IAM usuário `AdminUser` em sua AWS conta use o formato de ID mais longo para o tipo de `volume` recurso.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource volume \  
  --use-long-ids
```

O `modify-identity-id-format` exemplo a seguir permite que o IAM usuário `AdminUser` em sua AWS conta use o formato de ID mais longo para todos os tipos de recursos compatíveis que estão dentro do período de inscrição.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource all-current \  
  --use-long-ids
```

- Para API obter detalhes, consulte [ModifyIdentityIdFormat](#) na Referência de AWS CLI Comandos.

modify-image-attribute

O código de exemplo a seguir mostra como usar `modify-image-attribute`.

AWS CLI

Exemplo 1: Para tornar AMI público

O `modify-image-attribute` exemplo a seguir torna AMI público o especificado.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{Group=all}]"
```

Este comando não produz saída.

Exemplo 2: Para tornar um AMI privado

O `modify-image-attribute` exemplo a seguir torna o especificado AMI privado.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{Group=all}]"
```

Este comando não produz saída.

Exemplo 3: para conceder permissão de lançamento a uma AWS conta

O `modify-instance-attribute` exemplo a seguir concede permissões de lançamento para a AWS conta especificada.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Add=[{UserId=123456789012}]"
```

Este comando não produz saída.

Exemplo 4: Para remover a permissão de lançamento de uma AWS conta

O `modify-instance-attribute` exemplo a seguir remove as permissões de lançamento da AWS conta especificada.

```
aws ec2 modify-image-attribute \  
  --image-id ami-5731123e \  
  --launch-permission "Remove=[{UserId=123456789012}]"
```

- Para API obter detalhes, consulte [ModifyImageAttribute](#) na Referência de AWS CLI Comandos.

modify-instance-attribute

O código de exemplo a seguir mostra como usar `modify-instance-attribute`.

AWS CLI

Exemplo 1: Para modificar o tipo de instância

O `modify-instance-attribute` exemplo a seguir modifica o tipo de instância da instância especificada. A instância deve estar no estado `stopped`.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --instance-type '{"Value\': \'m1.small\'}"
```

Este comando não produz saída.

Exemplo 2: Para habilitar a rede avançada em uma instância

O `modify-instance-attribute` exemplo a seguir ativa a rede aprimorada para a instância especificada. A instância deve estar no estado `stopped`.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --sriov-net-support simple
```

Este comando não produz saída.

Exemplo 3: Para modificar o `sourceDestCheck` atributo

O `modify-instance-attribute` exemplo a seguir define o `sourceDestCheck` atributo da instância especificada como `true`. A instância deve estar em um VPC.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-  
check "{\"Value\": true}"
```

Este comando não produz saída.

Exemplo 4: Para modificar o `deleteOnTermination` atributo do volume raiz

O `modify-instance-attribute` exemplo a seguir define o `deleteOnTermination` atributo para o volume raiz da instância especificada com EBS suporte da Amazon como `false`. Por padrão, esse atributo é `true` para o volume raiz.

Comando:

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\", \"Ebs\":  
  {\"DeleteOnTermination\": false}}]"
```

Este comando não produz saída.

Exemplo 5: Para modificar os dados do usuário anexados a uma instância

O `modify-instance-attribute` exemplo a seguir adiciona o conteúdo do arquivo `UserData.txt` como `UserData` o da instância especificada.

Conteúdo do arquivo original `UserData.txt`:

```
#!/bin/bash
```

```
yum update -y
service httpd start
chkconfig httpd on
```

O conteúdo do arquivo deve ser codificado em base64. O primeiro comando converte o arquivo de texto em base64 e o salva como um novo arquivo.

Versão Linux/macOS do comando:

```
base64 UserData.txt > UserData.base64.txt
```

Este comando não produz saída.

Versão do comando para Windows:

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >
UserData.base64.txt
```

Saída:

```
Input Length = 67
Output Length = 152
CertUtil: -encode command completed successfully.
```

Agora você pode referenciar esse arquivo no CLI comando a seguir:

```
aws ec2 modify-instance-attribute \
  --instance-id=i-09b5a14dbca622e76 \
  --attribute userData --value file://UserData.base64.txt
```

Este comando não produz saída.

Para obter mais informações, consulte [Dados do usuário e AWS CLI](#) no Guia EC2 do usuário.

- Para API obter detalhes, consulte [ModifyInstanceAttribute](#) na Referência de AWS CLI Comandos.

modify-instance-capacity-reservation-attributes

O código de exemplo a seguir mostra como usar `modify-instance-capacity-reservation-attributes`.

AWS CLI

Exemplo 1: Para modificar as configurações de segmentação da reserva de capacidade de uma instância

O `modify-instance-capacity-reservation-attributes` exemplo a seguir modifica uma instância interrompida para atingir uma reserva de capacidade específica.

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification  
  'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

Saída:

```
{  
  "Return": true  
}
```

Exemplo 2: Para modificar as configurações de segmentação da reserva de capacidade de uma instância

O `modify-instance-capacity-reservation-attributes` exemplo a seguir modifica uma instância parada que tem como alvo a reserva de capacidade especificada para ser executada em qualquer reserva de capacidade que tenha atributos correspondentes (tipo de instância, plataforma, zona de disponibilidade) e que tenha critérios de correspondência de instâncias abertas.

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification 'CapacityReservationPreference=open'
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Modificar as configurações de reserva de capacidade de uma instância](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyInstanceCapacityReservationAttributes](#) na Referência de AWS CLI Comandos.

modify-instance-credit-specification

O código de exemplo a seguir mostra como usar `modify-instance-credit-specification`.

AWS CLI

Para modificar a opção de crédito para CPU uso de uma instância

Este exemplo modifica a opção de crédito para CPU uso da instância especificada na região especificada para “ilimitada”. As opções de crédito válidas são “padrão” e “ilimitadas”.

Comando:

```
aws ec2 modify-instance-credit-specification --instance-credit-specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

Saída:

```
{
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0"
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

- Para API obter detalhes, consulte [ModifyInstanceCreditSpecification](#) na Referência de AWS CLI Comandos.

modify-instance-event-start-time

O código de exemplo a seguir mostra como usar `modify-instance-event-start-time`.

AWS CLI

Para modificar o horário de início do evento para uma instância

O `modify-instance-event-start-time` comando a seguir mostra como modificar a hora de início do evento para a instância especificada. Especifique o ID do evento usando o `--instance-event-id` parâmetro. Especifique a nova data e hora usando o `--not-before` parâmetro.

```
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
```

Saída:

```
"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"
}
```

Para obter mais informações, consulte [Como trabalhar com instâncias programadas para reinicialização](#) no Guia do usuário do Amazon Elastic Compute Cloud

- Para API obter detalhes, consulte [ModifyInstanceEventStartTime](#) na Referência de AWS CLI Comandos.

modify-instance-event-window

O código de exemplo a seguir mostra como usar `modify-instance-event-window`.

AWS CLI

Exemplo 1: Para modificar o intervalo de tempo de uma janela de eventos

O `modify-instance-event-window` exemplo a seguir modifica o intervalo de tempo de uma janela de eventos. Especifique o `time-range` parâmetro para modificar o intervalo de tempo. Você também deve especificar o parâmetro `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
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 2: Para modificar um conjunto de intervalos de tempo para uma janela de eventos

O `modify-instance-event-window` exemplo a seguir modifica o intervalo de tempo de uma janela de eventos. Especifique o `time-range` parâmetro para modificar o intervalo de tempo. Você também deve especificar o parâmetro `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}]'
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```
}  
}
```

Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

Exemplo 3: Para modificar a expressão cron de uma janela de eventos

O `modify-instance-event-window` exemplo a seguir modifica a expressão cron de uma janela de eventos. Especifique o `cron-expression` para modificar a expressão cron. Você também deve especificar o parâmetro `time-range`.

```
aws ec2 modify-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --cron-expression "* 21-23 * * 2,3"
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```


Para restrições da janela de eventos, consulte [Considerações](#) na seção Eventos programados do Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [ModifyInstanceEventWindow](#) na Referência de AWS CLI Comandos.

modify-instance-maintenance-options

O código de exemplo a seguir mostra como usar `modify-instance-maintenance-options`.

AWS CLI

Exemplo 1: Para desativar o comportamento de recuperação de uma instância

O `modify-instance-maintenance-options` exemplo a seguir desativa a recuperação automática simplificada de uma instância em execução ou parada.

```
aws ec2 modify-instance-maintenance-options \  
  --instance-id i-0abcdef1234567890 \  
  --auto-recovery disabled
```

Saída:

```
{  
  "InstanceId": "i-0abcdef1234567890",  
  "AutoRecovery": "disabled"  
}
```

Para obter mais informações, consulte [Recupere sua instância](#) no Guia EC2 do usuário da Amazon para instâncias Linux.

Exemplo 2: Para definir o comportamento de recuperação de uma instância como padrão

O `modify-instance-maintenance-options` exemplo a seguir define o comportamento de recuperação automática como padrão, o que permite a recuperação automática simplificada para tipos de instância compatíveis.

```
aws ec2 modify-instance-maintenance-options \  
  --instance-id i-0abcdef1234567890 \  
  --auto-recovery default
```

Saída:

```
{
  "InstanceId": "i-0abcdef1234567890",
  "AutoRecovery": "default"
}
```

Para obter mais informações, consulte [Recupere sua instância](#) no Guia EC2 do usuário da Amazon para instâncias Linux.

- Para API obter detalhes, consulte [ModifyInstanceMaintenanceOptions](#) na Referência de AWS CLI Comandos.

modify-instance-metadata-options

O código de exemplo a seguir mostra como usar `modify-instance-metadata-options`.

AWS CLI

Exemplo 1: Para ativar IMDSv2

O `modify-instance-metadata-options` exemplo a seguir configura o uso de IMDSv2 na instância especificada.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-tokens required \
  --http-endpoint enabled
```

Saída:

```
{
  "InstanceId": "i-1234567898abcdef0",
  "InstanceMetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "enabled"
  }
}
```

Para obter mais informações, consulte [Metadados da instância e dados do usuário](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: para desativar os metadados da instância

O `modify-instance-metadata-options` exemplo a seguir desativa o uso de todas as versões dos metadados da instância na instância especificada.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-endpoint disabled
```

Saída:

```
{  
  "InstanceId": "i-1234567898abcdef0",  
  "InstanceMetadataOptions": {  
    "State": "pending",  
    "HttpTokens": "required",  
    "HttpPutResponseHopLimit": 1,  
    "HttpEndpoint": "disabled"  
  }  
}
```

Para obter mais informações, consulte [Metadados da instância e dados do usuário](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 3: Para habilitar o IPv6 endpoint de metadados da instância para sua instância

O `modify-instance-metadata-options` exemplo a seguir mostra como ativar o IPv6 endpoint para o serviço de metadados da instância.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-protocol-ipv6 enabled \  
  --http-endpoint enabled
```

Saída:

```
{  
  "InstanceId": "i-1234567898abcdef0",  
  "InstanceMetadataOptions": {  
    "State": "pending",  
    "HttpTokens": "required",  
    "HttpPutResponseHopLimit": 1,
```

```
    "HttpEndpoint": "enabled",  
    "HttpProtocolIpv6": "enabled"  
  }  
}
```

Por padrão, o IPv6 endpoint está desativado. Isso é verdade mesmo se você tiver iniciado uma instância em uma IPv6 sub-rede somente. O IPv6 endpoint IMDS só pode ser acessado em instâncias criadas no Sistema Nitro. Para obter mais informações, consulte [Metadados da instância e dados do usuário](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyInstanceMetadataOptions](#) na Referência de AWS CLI Comandos.

modify-instance-placement

O código de exemplo a seguir mostra como usar `modify-instance-placement`.

AWS CLI

Exemplo 1: Para remover a afinidade de uma instância com um host dedicado

O `modify-instance-placement` exemplo a seguir remove a afinidade de uma instância com um host dedicado e permite que ela seja executada em qualquer host dedicado disponível em sua conta que ofereça suporte ao tipo de instância.

```
aws ec2 modify-instance-placement \  
  --instance-id i-0e6ddf6187EXAMPLE \  
  --affinity default
```

Saída:

```
{  
  "Return": true  
}
```

Exemplo 2: Para estabelecer afinidade entre uma instância e o host dedicado especificado

O `modify-instance-placement` exemplo a seguir estabelece uma relação de lançamento entre uma instância e um host dedicado. A instância só pode ser executada no host dedicado especificado.

```
aws ec2 modify-instance-placement \  
  --instance-id i-0e6ddf6187EXAMPLE \  
  --affinity host \  
  --host-id i-0e6ddf6187EXAMPLE
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Modificar a localização e a afinidade da instância no Guia](#) do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 3: Para mover uma instância para um grupo de posicionamento

O `modify-instance-placement` exemplo a seguir move uma instância para um grupo de posicionamento, interrompe a instância, modifica o posicionamento da instância e, em seguida, reinicia a instância.

```
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
```

Para obter mais informações, consulte [Alterando o grupo de posicionamento de uma instância](#) no Amazon Elastic Compute Cloud Users Guide.

Exemplo 4: Para remover uma instância de um grupo de posicionamento

O `modify-instance-placement` exemplo a seguir remove uma instância de um grupo de posicionamento interrompendo a instância, modificando o posicionamento da instância e reiniciando a instância. O exemplo a seguir especifica uma string vazia (“”) para o nome do grupo de posicionamento para indicar que a instância não deve estar localizada em um grupo de posicionamento.

Pare a instância:

```
aws ec2 stop-instances \  
  --instance-ids i-0123a456700123456
```

Modifique o posicionamento (prompt de comando do Windows, Linux e macOS):

```
aws ec2 modify-instance-placement \  
  --instance-id i-0123a456700123456 \  
  --group-name ""
```

Modifique o posicionamento (Windows PowerShell):

```
aws ec2 modify-instance-placement `\  
  --instance-id i-0123a456700123456 `\  
  --group-name ""
```

Reinicie a instância:

```
aws ec2 start-instances \  
  --instance-ids i-0123a456700123456
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Modificar a localização e a afinidade da instância no Guia](#) do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [ModifyInstancePlacement](#) na Referência de AWS CLI Comandos.

modify-ipam-pool

O código de exemplo a seguir mostra como usar `modify-ipam-pool`.

AWS CLI

Para modificar um IPAM pool

O `modify-ipam-pool` exemplo a seguir modifica um 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
```

Saída:

```
{  
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Editar um pool](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [ModifyIpamPool](#) na Referência de AWS CLI Comandos.

modify-ipam-resource-cidr

O código de exemplo a seguir mostra como usar `modify-ipam-resource-cidr`.

AWS CLI

Para modificar o CIDR alocado para um recurso

O `modify-ipam-resource-cidr` exemplo a seguir modifica um recurso 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

```


Saída:

```
{
  "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"
  }
}
```

Para obter mais informações sobre movimentação de recursos, consulte [Mover recursos CIDRs entre escopos](#) no Guia do VPC IPAM usuário da Amazon.

Para obter mais informações sobre a alteração dos estados de monitoramento, consulte [Alterar o estado de monitoramento do recurso CIDRs](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [ModifyIpamResourceCidr](#) na Referência de AWS CLI Comandos.

modify-ipam-resource-discovery

O código de exemplo a seguir mostra como usar `modify-ipam-resource-discovery`.

AWS CLI

Para modificar as regiões operacionais de uma descoberta de recursos

Neste exemplo, você é um administrador IPAM delegado que deseja modificar as regiões operacionais de uma descoberta de recursos.

Para concluir essa solicitação:

Você não pode modificar uma descoberta de recursos padrão e deve ser o proprietário da descoberta do recurso. Você precisa da ID de descoberta do recurso, com a qual você pode obter. [describe-ipam-resource-discoveries](#)

O `modify-ipam-resource-discovery` exemplo a seguir modifica uma descoberta de recursos não padrão em sua AWS conta.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}  
}
```

Para obter mais informações, consulte [Trabalhe com descobertas de recursos](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [ModifyIpamResourceDiscovery](#) na Referência de AWS CLI Comandos.

modify-ipam-scope

O código de exemplo a seguir mostra como usar `modify-ipam-scope`.

AWS CLI

Para modificar a descrição de um escopo

Nesse cenário, você é um administrador IPAM delegado que deseja modificar a descrição de um IPAM escopo.

Para concluir essa solicitação, você precisará do ID do escopo, com o qual você pode obter [describe-ipam-scopes](#).

O `modify-ipam-scope` exemplo a seguir atualiza a descrição do escopo.

```
aws ec2 modify-ipam-scope \  
  --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \  
  --description example \  
  --region us-east-1
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações sobre escopos, consulte [Como IPAM funciona](#) no Guia do VPC IPAM usuário da Amazon.

- Para API obter detalhes, consulte [ModifyIpamScope](#) na Referência de AWS CLI Comandos.

modify-ipam

O código de exemplo a seguir mostra como usar `modify-ipam`.

AWS CLI

Para modificar um IPAM

O `modify-ipam` exemplo a seguir modifica um IPAM adicionando uma região operacional.

(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

```

Saída:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [ModifyIpam](#) na Referência de AWS CLI Comandos.

modify-launch-template

O código de exemplo a seguir mostra como usar `modify-launch-template`.

AWS CLI

Para alterar a versão padrão do modelo de lançamento

Este exemplo especifica a versão 2 do modelo de lançamento especificado como a versão padrão.

Comando:

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-version 2
```

Saída:

```
{
  "LaunchTemplate": {
```

```

    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "WebServers",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}

```

- Para API obter detalhes, consulte [ModifyLaunchTemplate](#) na Referência de AWS CLI Comandos.

modify-managed-prefix-list

O código de exemplo a seguir mostra como usar `modify-managed-prefix-list`.

AWS CLI

Para modificar uma lista de prefixos

O `modify-managed-prefix-list` exemplo a seguir adiciona uma entrada à lista de prefixos especificada.

```

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

```

Saída:

```

{
  "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"
  }
}

```

```
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [ModifyManagedPrefixList](#) na Referência de AWS CLI Comandos.

modify-network-interface-attribute

O código de exemplo a seguir mostra como usar `modify-network-interface-attribute`.

AWS CLI

Para modificar o atributo de anexo de uma interface de rede

Esse exemplo de comando modifica o `attachment` atributo da interface de rede especificada.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --  
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

Para modificar o atributo de descrição de uma interface de rede

Esse exemplo de comando modifica o `description` atributo da interface de rede especificada.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --  
description "My description"
```

Para modificar o `groupSet` atributo de uma interface de rede

Esse exemplo de comando modifica o `groupSet` atributo da interface de rede especificada.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --  
groups sg-903004f8 sg-1a2b3c4d
```

Para modificar o `sourceDestCheck` atributo de uma interface de rede

Esse exemplo de comando modifica o `sourceDestCheck` atributo da interface de rede especificada.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-source-dest-check
```

- Para API obter detalhes, consulte [ModifyNetworkInterfaceAttribute](#) na Referência de AWS CLI Comandos.

modify-private-dns-name-options

O código de exemplo a seguir mostra como usar `modify-private-dns-name-options`.

AWS CLI

Para modificar as opções, por exemplo, nomes de host

O `modify-private-dns-name-options` exemplo a seguir desativa a opção de responder a DNS consultas, por exemplo, nomes de host com registros A. DNS

```
aws ec2 modify-private-dns-name-options \  
  --instance-id i-1234567890abcdef0 \  
  --no-enable-resource-name-dns-a-record
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte os [tipos de nome de host de EC2 instância da Amazon](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [ModifyPrivateDnsNameOptions](#) na Referência de AWS CLI Comandos.

modify-reserved-instances

O código de exemplo a seguir mostra como usar `modify-reserved-instances`.

AWS CLI

Para modificar instâncias reservadas

Esse exemplo de comando move uma instância reservada para outra zona de disponibilidade na mesma região.

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
```

Saída:

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

Para modificar a plataforma de rede das Instâncias Reservadas

Este exemplo de comando converte EC2 -Classic Reserved Instances em -. 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
```

Saída:

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

Para obter mais informações, consulte [Modificar suas instâncias reservadas](#) no Guia do EC2 usuário da Amazon.

Para modificar o tamanho da instância das Instâncias Reservadas

Este exemplo de comando modifica uma instância reservada que tem 10 instâncias Linux/m1.small em us-west-1c para que 8 UNIX instâncias m1.small se tornem 2 instâncias m1.large

e as 2 m1.small restantes se tornem 1 instância m1.medium na mesma zona de disponibilidade.

Comando:

```
aws ec2 modify-reserved-instances --reserved-instances-ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=1,InstanceType=m1.medium
```

Saída:

```
{
  "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

Para obter mais informações, consulte Modificar o tamanho da instância de suas reservas no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [ModifyReservedInstances](#) na Referência de AWS CLI Comandos.

modify-security-group-rules

O código de exemplo a seguir mostra como usar modify-security-group-rules.

AWS CLI

Para modificar as regras de um grupo de segurança para atualizar a descrição da regra, o protocolo IP e o intervalo de CidrIpv 4 endereços

O modify-security-group-rules exemplo a seguir atualiza a descrição, o protocolo IP e o IPV4 CIDR intervalo de uma regra de grupo de segurança especificada. Use o security-group-rules parâmetro para inserir as atualizações para as regras do grupo de segurança especificado. -1 especifica todos os protocolos.

```
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}'
```

Saída:

```
{
  "Return": true
}
```

Para obter mais informações sobre regras de grupos de segurança, consulte [Regras de grupos de segurança](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [ModifySecurityGroupRules](#) na Referência de AWS CLI Comandos.

modify-snapshot-attribute

O código de exemplo a seguir mostra como usar `modify-snapshot-attribute`.

AWS CLI

Exemplo 1: Para modificar um atributo de snapshot

O `modify-snapshot-attribute` exemplo a seguir atualiza o `createVolumePermission` atributo do snapshot especificado, removendo as permissões de volume do usuário especificado.

```
aws ec2 modify-snapshot-attribute \
  --snapshot-id snap-1234567890abcdef0 \
  --attribute createVolumePermission \
  --operation-type remove \
  --user-ids 123456789012
```

Exemplo 2: Para tornar público um snapshot

O `modify-snapshot-attribute` exemplo a seguir torna público o snapshot especificado.

```
aws ec2 modify-snapshot-attribute \
  --snapshot-id snap-1234567890abcdef0 \
  --attribute createVolumePermission \
  --operation-type add \
  --group-names all
```

- Para API obter detalhes, consulte [ModifySnapshotAttribute](#) na Referência de AWS CLI Comandos.

modify-snapshot-tier

O código de exemplo a seguir mostra como usar `modify-snapshot-tier`.

AWS CLI

Exemplo 1: Para arquivar um instantâneo

O `modify-snapshot-tier` exemplo a seguir arquiva o instantâneo especificado.

```
aws ec2 modify-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --storage-tier archive
```

Saída:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "TieringStartTime": "2021-09-15T16:44:37.574Z"  
}
```

O parâmetro de `TieringStartTime` resposta indica a data e a hora em que o processo de arquivamento foi iniciado, no formato de UTC hora (YYYY-MM DDTHH -:MM:). SSZ

Para obter mais informações sobre arquivamento de instantâneos, consulte [Arquivar EBS instantâneos da Amazon no Guia](#) do usuário da Amazon EC2.

- Para API obter detalhes, consulte [ModifySnapshotTier](#) na Referência de AWS CLI Comandos.

modify-spot-fleet-request

O código de exemplo a seguir mostra como usar `modify-spot-fleet-request`.

AWS CLI

Para modificar uma solicitação de frota spot

Este exemplo de comando atualiza a capacidade alvo da solicitação de frota spot especificada.

Comando:

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-  
id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Saída:

```
{
  "Return": true
}
```

Esse exemplo de comando diminui a capacidade alvo da solicitação de frota spot especificada sem, como resultado, encerrar nenhuma instância 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
```

Saída:

```
{
  "Return": true
}
```

- Para API obter detalhes, consulte [ModifySpotFleetRequest](#) na Referência de AWS CLI Comandos.

modify-subnet-attribute

O código de exemplo a seguir mostra como usar `modify-subnet-attribute`.

AWS CLI

Para alterar o comportamento de IPv4 endereçamento público de uma sub-rede

Este exemplo modifica a `subnet-1a2b3c4d` para especificar que todas as instâncias executadas nessa sub-rede recebam um endereço público. IPv4 Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-
launch
```

Para alterar o comportamento de IPv6 endereçamento de uma sub-rede

Este exemplo modifica a subnet-1a2b3c4d para especificar que todas as instâncias executadas nessa sub-rede recebam um endereço do intervalo da sub-rede. IPv6

Comando:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

Para obter mais informações, consulte o Guia do usuário do endereçamento IP VPC em sua nuvem privada AWS virtual.

- Para API obter detalhes, consulte [ModifySubnetAttribute](#) na Referência de AWS CLI Comandos.

modify-traffic-mirror-filter-network-services

O código de exemplo a seguir mostra como usar modify-traffic-mirror-filter-network-services.

AWS CLI

Para adicionar serviços de rede a um filtro de espelho de tráfego

O modify-traffic-mirror-filter-network-services exemplo a seguir adiciona os serviços DNS de rede da Amazon ao filtro especificado.

```
aws ec2 modify-traffic-mirror-filter-network-services \
  --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \
  --add-network-service amazon-dns
```

Saída:

```
{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Modificar serviços de rede do filtro de espelhamento](#) de AWS tráfego no Guia de espelhamento de tráfego.

- Para API obter detalhes, consulte [ModifyTrafficMirrorFilterNetworkServices](#) na Referência de AWS CLI Comandos.

modify-traffic-mirror-filter-rule

O código de exemplo a seguir mostra como usar `modify-traffic-mirror-filter-rule`.

AWS CLI

Para modificar uma regra de filtro de espelhos de tráfego

O `modify-traffic-mirror-filter-rule` exemplo a seguir modifica a descrição da regra de filtro de espelhos de tráfego especificada.

```

aws ec2 modify-traffic-mirror-filter-rule \
  --traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \
  --description "TCP Rule"

```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Modificar suas regras de filtro de espelhos](#) de AWS tráfego no Guia de espelhamento de tráfego.

- Para API obter detalhes, consulte [ModifyTrafficMirrorFilterRule](#) na Referência de AWS CLI Comandos.

modify-traffic-mirror-session

O código de exemplo a seguir mostra como usar `modify-traffic-mirror-session`.

AWS CLI

Para modificar uma sessão de espelhamento de tráfego

O `modify-traffic-mirror-session` exemplo a seguir altera a descrição da sessão do espelhamento de tráfego e o número de pacotes a serem espelhados.

```
aws ec2 modify-traffic-mirror-session \
  --description "Change packet length" \
  --traffic-mirror-session-id tms-08a33b1214EXAMPLE \
  --remove-fields "packet-length"
```

Saída:

```
{
  "TrafficMirrorSession": {
    "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
    "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
```



```

    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
    "NetworkInterfaceId": "eni-070203f901EXAMPLE",
    "OwnerId": "111122223333",
    "SessionNumber": 1,
    "VirtualNetworkId": 7159709,
    "Description": "Change packet length",
    "Tags": []
  }
}

```

Para obter mais informações, consulte [Modificar sua sessão de espelhamento](#) de tráfego no Guia de espelhamento de tráfego.

- Para API obter detalhes, consulte [ModifyTrafficMirrorSession](#) na Referência de AWS CLI Comandos.

modify-transit-gateway-prefix-list-reference

O código de exemplo a seguir mostra como usar `modify-transit-gateway-prefix-list-reference`.

AWS CLI

Para modificar uma referência a uma lista de prefixos

O `modify-transit-gateway-prefix-list-reference` exemplo a seguir modifica a referência da lista de prefixos na tabela de rotas especificada alterando o anexo para o qual o tráfego é roteado.

```

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

```

Saída:

```

{
  "TransitGatewayPrefixListReference": {
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-11111122222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "modifying",
  }
}

```

```

    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}

```

Para obter mais informações, consulte as [referências da lista de prefixos](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [ModifyTransitGatewayPrefixListReferencena](#) Referência de AWS CLI Comandos.

modify-transit-gateway-vpc-attachment

O código de exemplo a seguir mostra como usar `modify-transit-gateway-vpc-attachment`.

AWS CLI

Para modificar um VPC anexo do Transit Gateway

O `modify-transit-gateway-vpc-attachment` exemplo a seguir adiciona uma sub-rede ao VPC anexo do gateway de trânsito especificado.

```

aws ec2 modify-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
  --add-subnet-ids subnet-0e51f45802EXAMPLE

```

Saída:

```

{
  "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"
    }
}
}

```

Para obter mais informações, consulte [Anexos do Transit Gateway a a VPC](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [ModifyTransitGatewayVpcAttachment](#) na Referência de AWS CLI Comandos.

modify-transit-gateway

O código de exemplo a seguir mostra como usar `modify-transit-gateway`.

AWS CLI

Para modificar um gateway de trânsito

O `modify-transit-gateway` exemplo a seguir modifica o gateway de trânsito especificado ativando o ECMP suporte para VPN anexos.

```

aws ec2 modify-transit-gateway \
  --transit-gateway-id tgw-111111222222aaaaa \
  --options VpnEcmpSupport=enable

```

Saída:

```

{
  "TransitGateway": {
    "TransitGatewayId": "tgw-111111222222aaaaa",
    "TransitGatewayArn": "64512",
    "State": "modifying",
    "OwnerId": "123456789012",
    "CreationTime": "2020-04-30T08:41:37.000Z",
    "Options": {
      "AmazonSideAsn": 64512,
      "AutoAcceptSharedAttachments": "disable",

```

```
        "DefaultRouteTableAssociation": "enable",
        "AssociationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
        "DefaultRouteTablePropagation": "enable",
        "PropagationDefaultRouteTableId": "tgw-rtb-0123456789abcd123",
        "VpnEcmpSupport": "enable",
        "DnsSupport": "enable"
    }
}
```

Para obter mais informações, consulte [Transit Gateways](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [ModifyTransitGateway](#) na Referência de AWS CLI Comandos.

modify-verified-access-endpoint-policy

O código de exemplo a seguir mostra como usar `modify-verified-access-endpoint-policy`.

AWS CLI

Para configurar a política de acesso verificado para um endpoint

O `modify-verified-access-endpoint-policy` exemplo a seguir adiciona a política de acesso verificado especificada ao endpoint de acesso verificado especificado.

```
aws ec2 modify-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --policy-enabled \
  --policy-document file://policy.txt
```

Conteúdo de `policy.txt`:

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

Saída:

```
{
  "PolicyEnabled": true,
```

```
"PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Para obter mais informações, consulte [Políticas de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [ModifyVerifiedAccessEndpointPolicy](#) na Referência de AWS CLI Comandos.

modify-verified-access-endpoint

O código de exemplo a seguir mostra como usar `modify-verified-access-endpoint`.

AWS CLI

Para modificar a configuração de um endpoint de acesso verificado

O `modify-verified-access-endpoint` exemplo a seguir adiciona a descrição especificada ao endpoint de acesso verificado especificado.

```
aws ec2 modify-verified-access-endpoint \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --description "Testing Verified Access"
```

Saída:

```
{
  "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"
}
}

```

Para obter mais informações, consulte [Endpoints de acesso verificado](#) no Guia do usuário do AWS Verified Access.

- Para API obter detalhes, consulte [ModifyVerifiedAccessEndpoint](#) na Referência de AWS CLI Comandos.

modify-verified-access-group-policy

O código de exemplo a seguir mostra como usar `modify-verified-access-group-policy`.

AWS CLI

Para configurar uma política de acesso verificado para um grupo

O `modify-verified-access-group-policy` exemplo a seguir adiciona a política de acesso verificado especificada ao grupo de acesso verificado especificado.

```

aws ec2 modify-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --policy-enabled \
  --policy-document file://policy.txt

```

Conteúdo de `policy.txt`:

```

permit(principal,action,resource)
when {

```

```
context.identity.groups.contains("finance") &&
context.identity.email.verified == true
};
```

Saída:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Para obter mais informações, consulte [Grupos de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [ModifyVerifiedAccessGroupPolicy](#) na Referência de AWS CLI Comandos.

modify-verified-access-group

O código de exemplo a seguir mostra como usar `modify-verified-access-group`.

AWS CLI

Para modificar a configuração de um grupo de acesso verificado

O `modify-verified-access-group` exemplo a seguir adiciona a descrição especificada ao grupo de acesso verificado especificado.

```
aws ec2 modify-verified-access-group \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --description "Testing Verified Access"
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Grupos de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [ModifyVerifiedAccessGroup](#) na Referência de AWS CLI Comandos.

modify-verified-access-instance-logging-configuration

O código de exemplo a seguir mostra como usar `modify-verified-access-instance-logging-configuration`.

AWS CLI

Para habilitar o registro em uma instância de acesso verificado

O `modify-verified-access-instance-logging-configuration` exemplo a seguir permite o registro de acesso para a instância de acesso verificado especificada. Os registros serão entregues ao grupo de CloudWatch registros de registros especificado.

```

aws ec2 modify-verified-access-instance-logging-configuration \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}

```

Saída:

```

{
  "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
}
}
}

```

Para obter mais informações, consulte [Registros de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [ModifyVerifiedAccessInstanceLoggingConfiguration](#) na Referência de AWS CLI Comandos.

modify-verified-access-instance

O código de exemplo a seguir mostra como usar `modify-verified-access-instance`.

AWS CLI

Para modificar a configuração de uma instância de acesso verificado

O `modify-verified-access-instance` exemplo a seguir adiciona a descrição especificada à instância de acesso verificado especificada.

```

aws ec2 modify-verified-access-instance \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --description "Testing Verified Access"

```

Saída:

```

{
  "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"
}
```

Para obter mais informações, consulte [Instâncias de acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [ModifyVerifiedAccessInstance](#) na Referência de AWS CLI Comandos.

modify-verified-access-trust-provider

O código de exemplo a seguir mostra como usar `modify-verified-access-trust-provider`.

AWS CLI

Para modificar a configuração de um provedor confiável de acesso verificado

O `modify-verified-access-trust-provider` exemplo a seguir adiciona a descrição especificada ao provedor confiável de acesso verificado especificado.

```
aws ec2 modify-verified-access-trust-provider \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
  --description "Testing Verified Access"
```

Saída:

```
{
  "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"  
  }  
}
```

Para obter mais informações, consulte [Provedores confiáveis para acesso verificado](#) no Guia do usuário de acesso AWS verificado.

- Para API obter detalhes, consulte [ModifyVerifiedAccessTrustProvider](#) na Referência de AWS CLI Comandos.

modify-volume-attribute

O código de exemplo a seguir mostra como usar `modify-volume-attribute`.

AWS CLI

Para modificar um atributo de volume

Este exemplo define o `autoEnableIo` atributo do volume com o ID `vol-1234567890abcdef0` como `true`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- Para API obter detalhes, consulte [ModifyVolumeAttribute](#) na Referência de AWS CLI Comandos.

modify-volume

O código de exemplo a seguir mostra como usar `modify-volume`.

AWS CLI

Exemplo 1: Para modificar um volume alterando seu tamanho

O `modify-volume` exemplo a seguir altera o tamanho do volume especificado para 150 GB.

Comando:

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

Saída:

```
{
  "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
  }
}
```

Exemplo 2: Para modificar um volume alterando seu tipo, tamanho e IOPS valor

O `modify-volume` exemplo a seguir altera o tipo de volume para Provisionado IOPSSSD, define a IOPS taxa alvo como 10000 e define o tamanho do volume como 350 GB.

```
aws ec2 modify-volume \
  --volume-type io1 \
  --iops 10000 \
  --size 350 \
  --volume-id vol-1234567890abcdef0
```

Saída:

```
{
  "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
  }
}
```

```
}  
}
```

- Para API obter detalhes, consulte [ModifyVolume](#) na Referência de AWS CLI Comandos.

modify-vpc-attribute

O código de exemplo a seguir mostra como usar `modify-vpc-attribute`.

AWS CLI

Para modificar o `enableDnsSupport` atributo

Este exemplo modifica o `enableDnsSupport` atributo. Esse atributo indica se a DNS resolução está habilitada para VPC o. Se esse atributo for `true`, o DNS servidor da Amazon resolverá os DNS nomes de host de suas instâncias para os endereços IP correspondentes; caso contrário, isso não acontece. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value  
\":false}"
```

Para modificar o `enableDnsHostnames` atributo

Este exemplo modifica o `enableDnsHostnames` atributo. Esse atributo indica se as instâncias foram executadas no VPC get DNS hostnames. Se esse atributo for `true`, instâncias em VPC get DNS hostnames; caso contrário, não o farão. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value  
\":false}"
```

- Para API obter detalhes, consulte [ModifyVpcAttribute](#) na Referência de AWS CLI Comandos.

modify-vpc-endpoint-connection-notification

O código de exemplo a seguir mostra como usar `modify-vpc-endpoint-connection-notification`.

AWS CLI

Para modificar uma notificação de conexão de endpoint

Este exemplo altera o SNS tópico da notificação de conexão do endpoint especificada.

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
```

Saída:

```
{
  "ReturnValue": true
}
```

- Para API obter detalhes, consulte [ModifyVpcEndpointConnectionNotification](#) na Referência de AWS CLI Comandos.

modify-vpc-endpoint-service-configuration

O código de exemplo a seguir mostra como usar `modify-vpc-endpoint-service-configuration`.

AWS CLI

Para modificar a configuração de um serviço de endpoint

Este exemplo altera o requisito de aceitação para o serviço de endpoint especificado.

Comando:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --no-acceptance-required
```

Saída:

```
{
  "ReturnValue": true
}
```

- Para API obter detalhes, consulte [ModifyVpcEndpointServiceConfiguration](#) na Referência de AWS CLI Comandos.

modify-vpc-endpoint-service-payer-responsibility

O código de exemplo a seguir mostra como usar `modify-vpc-endpoint-service-payer-responsibility`.

AWS CLI

Para modificar a responsabilidade do pagador

O `modify-vpc-endpoint-service-payer-responsibility` exemplo a seguir modifica a responsabilidade do pagador pelo serviço de endpoint especificado.

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \  
  --service-id vpce-svc-071afff70666e61e0 \  
  --payer-responsibility ServiceOwner
```

Este comando não produz saída.

- Para API obter detalhes, consulte [ModifyVpcEndpointServicePayerResponsibility](#) na Referência de AWS CLI Comandos.

modify-vpc-endpoint-service-permissions

O código de exemplo a seguir mostra como usar `modify-vpc-endpoint-service-permissions`.

AWS CLI

Para modificar as permissões do serviço de endpoint

Este exemplo adiciona permissão para que uma AWS conta se conecte ao serviço de endpoint especificado.

Comando:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

Saída:

```
{
  "ReturnValue": true
}
```

Este exemplo adiciona permissão para que um IAM usuário específico (admin) se conecte ao serviço de endpoint especificado.

Comando:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/  
admin"]'
```

- Para API obter detalhes, consulte [ModifyVpcEndpointServicePermissions](#) na Referência de AWS CLI Comandos.

modify-vpc-endpoint

O código de exemplo a seguir mostra como usar `modify-vpc-endpoint`.

AWS CLI

Para modificar um endpoint de gateway

Este exemplo modifica o endpoint do gateway `vpce-1a2b3c4d` associando a tabela `rtb-aaa222bb` de rotas ao endpoint e redefinindo o documento de política.

Comando:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-  
ids rtb-aaa222bb --reset-policy
```

Saída:

```
{
  "Return": true
}
```

Para modificar um endpoint de interface

Este exemplo modifica o endpoint da interface `vpce-0fe5b17a0707d6fa5` adicionando uma sub-rede `subnet-d6fcaa8d` ao endpoint.

Comando:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-id subnet-d6fcaa8d
```

Saída:

```
{  
  "Return": true  
}
```

- Para API obter detalhes, consulte [ModifyVpcEndpoint](#) na Referência de AWS CLI Comandos.

modify-vpc-peering-connection-options

O código de exemplo a seguir mostra como usar `modify-vpc-peering-connection-options`.

AWS CLI

Para habilitar a comunicação por meio de uma conexão VPC emparelhada a partir de sua conexão local ClassicLink

Neste exemplo, para conexão de emparelhamento `pcx-aaaabbbb`, o proprietário do solicitante VPC modifica as opções de conexão de VPC emparelhamento para permitir que uma ClassicLink conexão local se comunique com o peer. VPC

Comando:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

Saída:

```
{  
  "RequesterPeeringConnectionOptions": {  
    "AllowEgressFromLocalClassicLinkToRemoteVpc": true  
  }  
}
```

```
}  
}
```

Para habilitar a comunicação por meio de uma conexão VPC emparelhada de sua conexão local VPC para uma conexão remota ClassicLink

Neste exemplo, o proprietário do aceitante VPC modifica as opções de conexão de VPC emparelhamento para permitir que o local se comunique com VPC a ClassicLink conexão no peer. VPC

Comando:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-  
connection-id pcx-aaaabbbb --accepter-peering-connection-  
options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

Saída:

```
{  
  "AcceptorPeeringConnectionOptions": {  
    "AllowEgressFromLocalVpcToRemoteClassicLink": true  
  }  
}
```

Para ativar o suporte DNS de resolução para a conexão de VPC emparelhamento

Neste exemplo, o proprietário do solicitante VPC modifica as opções de conexão de VPC emparelhamento *pcx-aaaabbbb* para permitir que o local VPC resolva DNS nomes de host públicos em endereços IP privados quando consultado em instâncias no peer. VPC

Comando:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-  
aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

Saída:

```
{  
  "RequesterPeeringConnectionOptions": {  
    "AllowDnsResolutionFromRemoteVpc": true  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [ModifyVpcPeeringConnectionOptions](#) na Referência de AWS CLI Comandos.

modify-vpc-tenancy

O código de exemplo a seguir mostra como usar `modify-vpc-tenancy`.

AWS CLI

Para modificar a localização de um VPC

Este exemplo modifica a localização de VPC `vpc-1a2b3c4d` para `default`

Comando:

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

Saída:

```
{  
  "Return": true  
}
```

- Para API obter detalhes, consulte [ModifyVpcTenancy](#) na Referência de AWS CLI Comandos.

modify-vpn-connection-options

O código de exemplo a seguir mostra como usar `modify-vpn-connection-options`.

AWS CLI

Para modificar suas opções de VPN conexão

O `modify-vpn-connection-options` exemplo a seguir modifica o local IPv4 CIDR no lado do gateway do cliente da VPN conexão especificada.

```
aws ec2 modify-vpn-connection-options \  
  --vpn-connection-id vpn-1122334455aabbccd \  
  --local-ipv4-network-cidr 10.0.0.0/16
```

Saída:

```
{
  "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": ""
        }
      ]
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Modificar as opções de VPN conexão de site a site no Guia do usuário de site a AWS site. VPN](#)

- Para API obter detalhes, consulte [ModifyVpnConnectionOptions](#) na Referência de AWS CLI Comandos.

modify-vpn-connection

O código de exemplo a seguir mostra como usar `modify-vpn-connection`.

AWS CLI

Para modificar uma VPN conexão

O `modify-vpn-connection` exemplo a seguir altera o gateway de destino `vpn-12345678901234567` para VPN conexão com o gateway privado `virtualvgw-11223344556677889`:

```
aws ec2 modify-vpn-connection \  
  --vpn-connection-id vpn-12345678901234567 \  
  --vpn-gateway-id vgw-11223344556677889
```

Saída:

```
{  
  "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"
  }
]
}
}

```

- Para API obter detalhes, consulte [ModifyVpnConnection](#) na Referência de AWS CLI Comandos.

modify-vpn-tunnel-certificate

O código de exemplo a seguir mostra como usar `modify-vpn-tunnel-certificate`.

AWS CLI

Para girar um certificado de VPN túnel

O `modify-vpn-tunnel-certificate` exemplo a seguir gira o certificado para o túnel especificado para uma VPN conexão.

```

aws ec2 modify-vpn-tunnel-certificate \
  --vpn-tunnel-outside-ip-address 203.0.113.17 \
  --vpn-connection-id vpn-12345678901234567

```

Saída:

```

{
  "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"
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ModifyVpnTunnelCertificate](#) na Referência de AWS CLI Comandos.

modify-vpn-tunnel-options

O código de exemplo a seguir mostra como usar `modify-vpn-tunnel-options`.

AWS CLI

Para modificar as opções de túnel para uma VPN conexão

O `modify-vpn-tunnel-options` exemplo a seguir atualiza os grupos Diffie-Hellman permitidos para o túnel e a conexão especificados. 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}]
```

Saída:

```
{  
  "VpnConnection": {  
    "CustomerGatewayConfiguration": "...configuration information...",  
    "CustomerGatewayId": "cgw-aabbccdee1122334",  
    "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"
    }
]
}
}
```

- Para API obter detalhes, consulte [ModifyVpnTunnelOptions](#) na Referência de AWS CLI Comandos.

monitor-instances

O código de exemplo a seguir mostra como usar `monitor-instances`.

AWS CLI

Para habilitar o monitoramento detalhado de uma instância

Este exemplo de comando habilita o monitoramento detalhado da instância especificada.

Comando:

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

Saída:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "pending"
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [MonitorInstances](#) na Referência de AWS CLI Comandos.

move-address-to-vpc

O código de exemplo a seguir mostra como usar `move-address-to-vpc`.

AWS CLI

Para mover um endereço para EC2 - VPC

Este exemplo move o endereço IP elástico 54.123.4.56 para a plataforma -. EC2 VPC

Comando:

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

Saída:

```
{
  "Status": "MoveInProgress"
}
```

- Para API obter detalhes, consulte [MoveAddressToVpc](#) na Referência de AWS CLI Comandos.

move-byoip-cidr-to-ipam

O código de exemplo a seguir mostra como usar `move-byoip-cidr-to-ipam`.

AWS CLI

Para transferir um BYOIP CIDR para IPAM

O `move-byoip-cidr-to-ipam` exemplo a seguir transfere um BYOIP CIDR para 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
```

Saída:

```
{
  "ByoipCidr": {
    "Cidr": "130.137.249.0/24",
    "State": "pending-transfer"
  }
}
```

```
}
```

Para obter mais informações, consulte [Tutorial: Transferir um BYOIP IPv4 CIDR para existente IPAM](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [MoveByoipCidrToIpam](#) na Referência de AWS CLI Comandos.

network-insights-access-scope

O código de exemplo a seguir mostra como usar `network-insights-access-scope`.

AWS CLI

Para criar escopos de acesso do Network Insights

O `create-network-insights-access-scope` exemplo a seguir cria um escopo de acesso ao Network Insights em sua AWS conta.

```
aws ec2 create-network-insights-access-scope \  
  --cli-input-json file://access-scope-file.json
```

Conteúdo de `access-scope-file.json`:

```
{  
  {  
    "MatchPaths": [  
      {  
        "Source": {  
          "ResourceStatement": {  
            "Resources": [  
              "vpc-abcd12e3"  
            ]  
          }  
        }  
      ]  
    },  
    "ExcludePaths": [  
      {  
        "Source": {  
          "ResourceStatement": {  
            "ResourceTypes": [  

```



```
    }
  ]
}
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI](#) Network Access Analyzer.

- Para API obter detalhes, consulte [NetworkInsightsAccessScopena](#) Referência de AWS CLI Comandos.

provision-byoip-cidr

O código de exemplo a seguir mostra como usar `provision-byoip-cidr`.

AWS CLI

Para provisionar um intervalo de endereços

O `provision-byoip-cidr` exemplo a seguir provisiona um intervalo de endereços IP públicos para uso com AWS.

```
aws ec2 provision-byoip-cidr \
  --cidr 203.0.113.25/24 \
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Saída:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "State": "pending-provision"
  }
}
```

Para obter mais informações sobre a criação de cadeias de mensagens para o contexto de autorização, consulte [Traga seus próprios endereços IP](#) no Guia do EC2 usuário da Amazon.

- Para API obter detalhes, consulte [ProvisionByoipCidr](#)na Referência de AWS CLI Comandos.

provision-ipam-pool-cidr

O código de exemplo a seguir mostra como usar `provision-ipam-pool-cidr`.

AWS CLI

Para provisionar CIDR a para um IPAM pool

O `provision-ipam-pool-cidr` exemplo a seguir provisiona a para um 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
```

Saída:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "10.0.0.0/24",  
    "State": "pending-provision"  
  }  
}
```

Para obter mais informações, consulte [CIDRs Provisionar para um pool](#) no Guia VPC IPAM do usuário da Amazon.

- Para API obter detalhes, consulte [ProvisionIpamPoolCidr](#) na Referência de AWS CLI Comandos.

purchase-host-reservation

O código de exemplo a seguir mostra como usar `purchase-host-reservation`.

AWS CLI

Para comprar uma reserva de anfitrião dedicado

Este exemplo compra a oferta de reserva de host dedicado especificada para o host dedicado especificado em sua conta.

Comando:

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [PurchaseHostReservation](#) na Referência de AWS CLI Comandos.

purchase-reserved-instances-offering

O código de exemplo a seguir mostra como usar `purchase-reserved-instances-offering`.

AWS CLI

Para comprar uma oferta de instância reservada

Este exemplo de comando ilustra a compra de uma oferta de Instâncias Reservadas, especificando o ID da oferta e a contagem de instâncias.

Comando:

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

Saída:

```
{
  "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
}
```

- Para API obter detalhes, consulte [PurchaseReservedInstancesOffering](#) na Referência de AWS CLI Comandos.

purchase-scheduled-instances

O código de exemplo a seguir mostra como usar `purchase-scheduled-instances`.

AWS CLI

Para comprar uma instância programada

Este exemplo compra uma instância programada.

Comando:

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

Purchase-Request.json:

```
[
  {
    "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
    "InstanceCount": 1
  }
]
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [PurchaseScheduledInstances](#) na Referência de AWS CLI Comandos.

reboot-instances

O código de exemplo a seguir mostra como usar `reboot-instances`.

AWS CLI

Para reinicializar uma instância da Amazon EC2

Este exemplo reinicia a instância especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

Para obter mais informações, consulte Reinicializar a instância no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [RebootInstances](#) na Referência de AWS CLI Comandos.

register-image

O código de exemplo a seguir mostra como usar `register-image`.

AWS CLI

Exemplo 1: Para registrar e AMI usar um arquivo de manifesto

O `register-image` exemplo a seguir registra e AMI usa o arquivo de manifesto especificado no Amazon S3.

```
aws ec2 register-image \  
  --name my-image \  
  --image-location my-s3-bucket/myimage/image.manifest.xml
```

Saída:

```
{  
  "ImageId": "ami-1234567890EXAMPLE"  
}
```

Para obter mais informações, consulte [Amazon Machine Images \(AMI\)](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para registrar um AMI usando um instantâneo de um dispositivo raiz

O `register-image` exemplo a seguir registra e AMI usa o snapshot especificado de um volume EBS raiz como dispositivo. `/dev/xvda` O mapeamento do dispositivo de bloco também inclui um EBS volume vazio de 100 GiB como 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}
```

Saída:

```
{  
  "ImageId": "ami-1a2b3c4d5eEXAMPLE"  
}
```

Para obter mais informações, consulte [Amazon Machine Images \(AMI\)](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [RegisterImage](#) na Referência de AWS CLI Comandos.

register-instance-event-notification-attributes

O código de exemplo a seguir mostra como usar `register-instance-event-notification-attributes`.

AWS CLI

Exemplo 1: Para incluir todas as tags nas notificações de eventos

O `register-instance-event-notification-attributes` exemplo a seguir inclui todas as tags nas notificações de eventos.

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute IncludeAllTagsOfInstance=true
```

Saída:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [],  
    "IncludeAllTagsOfInstance": true  
  }  
}
```

Para obter mais informações, consulte [Eventos programados para suas instâncias](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

Exemplo 2: Para incluir tags específicas nas notificações de eventos

O `register-instance-event-notification-attributes` exemplo a seguir inclui as tags especificadas nas notificações de eventos. Você não pode especificar tags se `IncludeAllTagsOfInstance` estiver `true`.

```
aws ec2 register-instance-event-notification-attributes \
  --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

Saída:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [
      "tag-key1",
      "tag-key2"
    ],
    "IncludeAllTagsOfInstance": false
  }
}
```

Para obter mais informações, consulte [Eventos programados para suas instâncias](#) no Guia do usuário do Amazon Elastic Compute Cloud para instâncias Linux.

- Para API obter detalhes, consulte [RegisterInstanceEventNotificationAttributes](#) na Referência de AWS CLI Comandos.

register-transit-gateway-multicast-group-sources

O código de exemplo a seguir mostra como usar `register-transit-gateway-multicast-group-sources`.

AWS CLI

Para registrar uma origem em um grupo multicast do Transit Gateway.

O `register-transit-gateway-multicast-group-sources` exemplo a seguir registra a origem do grupo de interface de rede especificado com um grupo 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
```

Saída:

```
{  
  "RegisteredMulticastGroupSources": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",  
    "RegisteredNetworkInterfaceIds": [  
      "eni-07f290fc3c090cbae"  
    ],  
    "GroupIpAddress": "224.0.1.0"  
  }  
}
```

Para obter mais informações, consulte [Registrar fontes com um grupo multicast no Guia](#) do usuário do AWS Transit Gateways.

- Para API obter detalhes, consulte [RegisterTransitGatewayMulticastGroupSources](#) na Referência de AWS CLI Comandos.

register-transit-gateway-multicast-group-members

O código de exemplo a seguir mostra como usar `register-transit-gateway-multicast-group-members`.

AWS CLI

Para ver as informações sobre as associações de domínio multicast do Transit Gateway

O `register-transit-gateway-multicast-group-members` exemplo a seguir retorna as associações para o domínio multicast especificado.

```
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
```

Saída:

```
{
  "RegisteredMulticastGroupMembers": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-0e246d32695012e81"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Guia do usuário do Transit Gateways.

- Para API obter detalhes, consulte [RegisterTransitGatewayMulticastGroupMembers](#) na Referência de AWS CLI Comandos.

register-transit-gateway-multicast-group-sources

O código de exemplo a seguir mostra como usar `register-transit-gateway-multicast-group-sources`.

AWS CLI

Para registrar uma origem em um grupo multicast do Transit Gateway.

O `register-transit-gateway-multicast-group-sources` exemplo a seguir registra a origem do grupo de interface de rede especificado com um grupo 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
```

Saída:

```
{
  "RegisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
  },
}
```

```

    "GroupIpAddress": "224.0.1.0"
  }
}

```

Para obter mais informações, consulte [Gerenciando domínios multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [RegisterTransitGatewayMulticastGroupSources](#) na Referência de AWS CLI Comandos.

reject-transit-gateway-peering-attachment

O código de exemplo a seguir mostra como usar `reject-transit-gateway-peering-attachment`.

AWS CLI

Para rejeitar um anexo de emparelhamento do Transit Gateway

O `reject-transit-gateway-peering-attachment` exemplo a seguir rejeita a solicitação de anexo de emparelhamento do Transit Gateway especificada. O `--region` parâmetro especifica a região na qual o gateway de trânsito aceitante está localizado.

```

aws ec2 reject-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
  --region us-east-2

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Transit Gateway Peering Attachments](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [RejectTransitGatewayPeeringAttachment](#) na Referência de AWS CLI Comandos.

reject-transit-gateway-vpc-attachment

O código de exemplo a seguir mostra como usar `reject-transit-gateway-vpc-attachment`.

AWS CLI

Para rejeitar um VPC anexo do Transit Gateway

O `reject-transit-gateway-vpc-attachment` exemplo a seguir rejeita o VPC anexo do gateway de trânsito especificado.

```

aws ec2 reject-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE

```

Saída:

```

{
  "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"
    }
  }
}

```

```
}
}
```

Para obter mais informações, consulte [Anexos do Transit Gateway a a VPC](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [RejectTransitGatewayVpcAttachment](#) na Referência de AWS CLI Comandos.

reject-transit-gateway-vpc-attachments

O código de exemplo a seguir mostra como usar `reject-transit-gateway-vpc-attachments`.

AWS CLI

Para rejeitar um VPC anexo do Transit Gateway

O `reject-transit-gateway-vpc-attachment` exemplo a seguir rejeita o VPC anexo do gateway de trânsito especificado.

```
aws ec2 reject-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Anexos do Transit Gateway a a VPC](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [RejectTransitGatewayVpcAttachments](#) na Referência de AWS CLI Comandos.

reject-vpc-endpoint-connections

O código de exemplo a seguir mostra como usar `reject-vpc-endpoint-connections`.

AWS CLI

Para rejeitar uma solicitação de conexão de endpoint de interface

Este exemplo rejeita a solicitação de conexão de endpoint especificada para o serviço de endpoint especificado.

Comando:

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --  
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Saída:

```
{  
  "Unsuccessful": []  
}
```

- Para API obter detalhes, consulte [RejectVpcEndpointConnections](#) na Referência de AWS CLI Comandos.

reject-vpc-peering-connection

O código de exemplo a seguir mostra como usar `reject-vpc-peering-connection`.

AWS CLI

Para rejeitar uma conexão de VPC emparelhamento

Este exemplo rejeita a solicitação de conexão de VPC emparelhamento especificada.

Comando:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Saída:

```
{  
  "Return": true  
}
```

- Para API obter detalhes, consulte [RejectVpcPeeringConnection](#) na Referência de AWS CLI Comandos.

release-address

O código de exemplo a seguir mostra como usar `release-address`.

AWS CLI

Para liberar um endereço IP elástico para EC2 -Classic

Este exemplo libera um endereço IP elástico para uso com instâncias em EC2 -Classic. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 release-address --public-ip 198.51.100.0
```

Para liberar um endereço IP elástico para EC2 - VPC

Este exemplo libera um endereço IP elástico para uso com instâncias em umVPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- Para API obter detalhes, consulte [ReleaseAddress](#) na Referência de AWS CLI Comandos.

release-hosts

O código de exemplo a seguir mostra como usar `release-hosts`.

AWS CLI

Para liberar um host dedicado da sua conta

Para liberar um host dedicado da sua conta. As instâncias que estão no host devem ser interrompidas ou encerradas antes que o host possa ser liberado.

Comando:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

Saída:

```
{
  "Successful": [
    "h-0029d6e3cacf1b3da"
  ],
  "Unsuccessful": []
}
```

- Para API obter detalhes, consulte [ReleaseHosts](#) na Referência de AWS CLI Comandos.

release-ipam-pool-allocation

O código de exemplo a seguir mostra como usar `release-ipam-pool-allocation`.

AWS CLI

Para liberar uma alocação de IPAM pool

Neste exemplo, você é um administrador IPAM delegado que tentou excluir um IPAM pool, mas recebeu um erro informando que você não pode excluir o pool enquanto o pool tem alocações. Você está usando esse comando para liberar uma alocação de pool.

Observe o seguinte:

Você só pode usar esse comando para alocações personalizadas. Para remover uma alocação para um recurso sem excluir o recurso, defina seu estado monitorado como falso usando [modify-ipam-resource-cidr](#). Para concluir essa solicitação, você precisará do ID do IPAM pool, com o

qual você pode obter. [describe-ipam-pools](#) Você também precisará do ID de alocação, que pode ser obtido com [get-ipam-pool-allocations](#). Se não quiser remover as alocações uma por uma, você pode usar o `--cascade` option ao excluir um IPAM pool para liberar automaticamente todas as alocações no pool antes de excluí-las. Há vários pré-requisitos antes de executar esse comando. Para obter mais informações, consulte [Liberar uma alocação](#) no Guia VPC IPAM do usuário da Amazon. O local `--region` em que você executa esse comando deve ser o local do IPAM pool em que a alocação está.

O `release-ipam-pool-allocation` exemplo a seguir libera uma alocação de IPAM 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
```

Saída:

```
{
  "Success": true
}
```

Depois de liberar uma alocação, talvez você queira [delete-ipam-pool](#) executá-la.

- Para API obter detalhes, consulte [ReleaseIpamPoolAllocation](#) na Referência de AWS CLI Comandos.

replace-iam-instance-profile-association

O código de exemplo a seguir mostra como usar `replace-iam-instance-profile-association`.

AWS CLI

Para substituir um perfil de IAM instância por uma instância

Este exemplo substitui o perfil da IAM instância representado pela associação `iip-assoc-060bae234aac2e7fa` pelo perfil da IAM instância chamado `AdminRole`.

```
aws ec2 replace-iam-instance-profile-association \
```

```
--iam-instance-profile Name=AdminRole \  
--association-id iip-assoc-060bae234aac2e7fa
```

Saída:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-087711ddaf98f9489",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0b215292fab192820",  
    "IamInstanceProfile": {  
      "Id": "AIPAJLNLDX3AMYZNWYYAY",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [ReplacelamInstanceProfileAssociation](#) na Referência de AWS CLI Comandos.

replace-network-acl-association

O código de exemplo a seguir mostra como usar `replace-network-acl-association`.

AWS CLI

Para substituir a rede ACL associada a uma sub-rede

Este exemplo associa a rede especificada ACL à sub-rede da associação de rede ACL especificada.

Comando:

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --  
network-acl-id acl-5fb85d36
```

Saída:

```
{  
  "NewAssociationId": "aclassoc-3999875b"
```

```
}
```

- Para API obter detalhes, consulte [ReplaceNetworkAclAssociation](#) na Referência de AWS CLI Comandos.

replace-network-acl-entry

O código de exemplo a seguir mostra como usar `replace-network-acl-entry`.

AWS CLI

Para substituir uma ACL entrada de rede

Este exemplo substitui uma entrada para a rede ACL especificada. A nova regra 100 permite o tráfego de entrada de 203.0.113.12/24 na UDP porta 53 () em qualquer sub-rede associada. 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
```

- Para API obter detalhes, consulte [ReplaceNetworkAclEntry](#) na Referência de AWS CLI Comandos.

replace-route-table-association

O código de exemplo a seguir mostra como usar `replace-route-table-association`.

AWS CLI

Para substituir a tabela de rotas associada a uma sub-rede

Este exemplo associa a tabela de rotas especificada à sub-rede para a associação da tabela de rotas especificada.

Comando:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```


Saída:

```
{
  "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- Para API obter detalhes, consulte [ReplaceRouteTableAssociation](#) na Referência de AWS CLI Comandos.

replace-route

O código de exemplo a seguir mostra como usar `replace-route`.

AWS CLI

Para substituir uma rota

Este exemplo substitui a rota especificada na tabela de rotas especificada. A nova rota corresponde à especificada CIDR e envia o tráfego para o gateway privado virtual especificado. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-  
block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- Para API obter detalhes, consulte [ReplaceRoute](#) na Referência de AWS CLI Comandos.

replace-transit-gateway-route

O código de exemplo a seguir mostra como usar `replace-transit-gateway-route`.

AWS CLI

Para substituir a rota especificada na tabela de rotas do gateway de trânsito especificada

O `replace-transit-gateway-route` exemplo a seguir substitui a rota na tabela de rotas do gateway de trânsito especificada.

```
aws ec2 replace-transit-gateway-route \  

```

```
--destination-cidr-block 10.0.2.0/24 \  
--transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE \  
--transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

Saída:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  
      {  
        "ResourceId": "vpc-4EXAMPLE",  
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
        "ResourceType": "vpc"  
      }  
    ],  
    "Type": "static",  
    "State": "active"  
  }  
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [ReplaceTransitGatewayRoute](#) na Referência de AWS CLI Comandos.

report-instance-status

O código de exemplo a seguir mostra como usar `report-instance-status`.

AWS CLI

Para relatar feedback de status de uma instância

Este exemplo de comando relata o feedback de status da instância especificada.

Comando:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --  
reason-codes unresponsive
```

- Para API obter detalhes, consulte [ReportInstanceStatus](#) na Referência de AWS CLI Comandos.

request-spot-fleet

O código de exemplo a seguir mostra como usar `request-spot-fleet`.

AWS CLI

Para solicitar uma frota spot na sub-rede com o menor preço

Esse exemplo de comando cria uma solicitação de frota Spot com duas especificações de lançamento que diferem somente por sub-rede. A frota Spot executa as instâncias na sub-rede especificada com o menor preço. Se as instâncias forem iniciadas em um padrãoVPC, elas receberão um endereço IP público por padrão. Se as instâncias forem executadas de forma não padrãoVPC, elas não receberão um endereço IP público por padrão.

Observe que você não pode especificar sub-redes diferentes da mesma zona de disponibilidade em uma solicitação de frota 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"
      }
    }
  ]
}
```

```
    }  
  }  
]  
}
```

Saída:

```
{  
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

Para solicitar uma frota spot na zona de disponibilidade com o menor preço

Esse exemplo de comando cria uma solicitação de frota spot com duas especificações de lançamento que diferem somente por zona de disponibilidade. A frota spot lança as instâncias na zona de disponibilidade especificada com o menor preço. Se sua conta suportar VPC apenas EC2 -, a Amazon EC2 executa as instâncias spot na sub-rede padrão da zona de disponibilidade. Se sua conta suportar EC2 -Classic, a Amazon EC2 iniciará as instâncias em EC2 -Classic na zona de disponibilidade.

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"
    }
}
]
```

Para iniciar instâncias spot em uma sub-rede e atribuir a elas endereços IP públicos

Este exemplo de comando atribui endereços públicos às instâncias executadas em um formato não VPC padrão. Observe que, ao especificar uma interface de rede, você deve incluir a ID da sub-rede e a ID do grupo de segurança usando a interface de rede.

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"
      }
    }
  ]
}
```

```
}
```

Para solicitar uma frota Spot usando a estratégia de alocação diversificada

Esse exemplo de comando cria uma solicitação de frota spot que executa 30 instâncias usando a estratégia de alocação diversificada. As especificações de lançamento variam de acordo com o tipo de instância. A frota Spot distribui as instâncias de acordo com as especificações de lançamento, de forma que haja 10 instâncias de cada 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"
    }
  ]
}
```

Para obter mais informações, consulte Solicitações de frota spot no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [RequestSpotFleet](#) na Referência de AWS CLI Comandos.

request-spot-instances

O código de exemplo a seguir mostra como usar `request-spot-instances`.

AWS CLI

Para solicitar instâncias spot

Este exemplo de comando cria uma solicitação única de instância spot para cinco instâncias na zona de disponibilidade especificada. Se sua conta suportar VPC apenas EC2 -, a Amazon EC2 executa as instâncias na sub-rede padrão da zona de disponibilidade especificada. Se sua conta suportar EC2 -Classic, a Amazon EC2 executará as instâncias em EC2 -Classic na zona de disponibilidade especificada.

Comando:

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Especificação.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"
  }
}
```

Saída:

```
{
  "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"
},
...
]
}

```

Este exemplo de comando cria uma solicitação única de instância spot para cinco instâncias na sub-rede especificada. A Amazon EC2 lança as instâncias na sub-rede especificada. Se não VPC for padrãoVPC, as instâncias não receberão um endereço IP público por padrão.

Comando:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```


Especificação.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

Saída:

```
{
  "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"
},
...
]
}

```

Este exemplo atribui um endereço IP público às Instâncias Spot que você executa de forma não VPC padrão. Observe que, ao especificar uma interface de rede, você deve incluir a ID da sub-rede e a ID do grupo de segurança usando a interface de rede.

Comando:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

Especificação.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"
  }
}

```

- Para API obter detalhes, consulte [RequestSpotInstances](#) na Referência de AWS CLI Comandos.

reset-address-attribute

O código de exemplo a seguir mostra como usar `reset-address-attribute`.

AWS CLI

Para redefinir o atributo do nome de domínio associado a um endereço IP elástico

Os `reset-address-attribute` exemplos a seguir redefinem o atributo de nome de domínio de um endereço IP elástico.

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
```

Saída:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.com."  
      "PtrRecordUpdate": {  
        "Value": "example.net.",  
        "Status": "PENDING"  
      }  
    }  
  ]  
}
```

Para monitorar a alteração pendente, consulte [describe-addresses-attribute](#) na Referência de AWS CLI Comandos.

- Para API obter detalhes, consulte [ResetAddressAttribute](#) na Referência de AWS CLI Comandos.

reset-ebs-default-kms-key-id

O código de exemplo a seguir mostra como usar `reset-ebs-default-kms-key-id`.

AWS CLI

Para redefinir seu padrão CMK de EBS criptografia

O `reset-ebs-default-kms-key-id` exemplo a seguir redefine o padrão CMK de EBS criptografia da sua AWS conta na região atual.

```
aws ec2 reset-ebs-default-kms-key-id
```

Saída:

```
{
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513eEXAMPLE"
}
```

- Para API obter detalhes, consulte [ResetEbsDefaultKmsKeyId](#) na Referência de AWS CLI Comandos.

reset-fpga-image-attribute

O código de exemplo a seguir mostra como usar `reset-fpga-image-attribute`.

AWS CLI

Para redefinir os atributos de uma FPGA imagem da Amazon

Este exemplo redefine as permissões de carregamento para o especificadoAFI.

Comando:

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

Saída:

```
{
```

```
"Return": true
}
```

- Para API obter detalhes, consulte [ResetFpgaImageAttribute](#) na Referência de AWS CLI Comandos.

reset-image-attribute

O código de exemplo a seguir mostra como usar `reset-image-attribute`.

AWS CLI

Para redefinir o `launchPermission` atributo

Este exemplo redefine o `launchPermission` atributo do especificado AMI para seu valor padrão. Por padrão, AMIs são privados. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute LaunchPermission
```

- Para API obter detalhes, consulte [ResetImageAttribute](#) na Referência de AWS CLI Comandos.

reset-instance-attribute

O código de exemplo a seguir mostra como usar `reset-instance-attribute`.

AWS CLI

Para redefinir o `sourceDestCheck` atributo

Este exemplo redefine o `sourceDestCheck` atributo da instância especificada. A instância deve estar em um VPC. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute sourceDestCheck
```

Para redefinir o atributo do kernel

Este exemplo redefine o `kernel` atributo da instância especificada. A instância deve estar no estado `stopped`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute kernel
```

Para redefinir o atributo `ramdisk`

Este exemplo redefine o `ramdisk` atributo da instância especificada. A instância deve estar no estado `stopped`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute ramdisk
```

- Para API obter detalhes, consulte [ResetInstanceAttribute](#) na Referência de AWS CLI Comandos.

reset-network-interface-attribute

O código de exemplo a seguir mostra como usar `reset-network-interface-attribute`.

AWS CLI

Para redefinir um atributo de interface de rede

O `reset-network-interface-attribute` exemplo a seguir redefine o valor do atributo de verificação de origem/destino para `true`

```
aws ec2 reset-network-interface-attribute \  
--network-interface-id eni-686ea200 \  
--source-dest-check
```

Este comando não produz saída.

- Para API obter detalhes, consulte [ResetNetworkInterfaceAttribute](#) na Referência de AWS CLI Comandos.

reset-snapshot-attribute

O código de exemplo a seguir mostra como usar `reset-snapshot-attribute`.

AWS CLI

Para redefinir um atributo de snapshot

Este exemplo redefine as permissões de criação de volume para `snapshotsnap-1234567890abcdef0`. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --  
attribute createVolumePermission
```

- Para API obter detalhes, consulte [ResetSnapshotAttribute](#) na Referência de AWS CLI Comandos.

restore-address-to-classic

O código de exemplo a seguir mostra como usar `restore-address-to-classic`.

AWS CLI

Para restaurar um endereço para EC2 -Classic

Este exemplo restaura o endereço IP elástico `198.51.100.0` para a plataforma -Classic. EC2

Comando:

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

Saída:

```
{  
  "Status": "MoveInProgress",  
  "PublicIp": "198.51.100.0"  
}
```

- Para API obter detalhes, consulte [RestoreAddressToClassic](#) na Referência de AWS CLI Comandos.

restore-image-from-recycle-bin

O código de exemplo a seguir mostra como usar `restore-image-from-recycle-bin`.

AWS CLI

Para restaurar uma imagem da Lixeira

O `restore-image-from-recycle-bin` exemplo a seguir restaura AMI `ami-0111222333444abcd` da Lixeira.

```
aws ec2 restore-image-from-recycle-bin \  
  --image-id ami-0111222333444abcd
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Recuperar AMIs da lixeira](#) no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [RestoreImageFromRecycleBin](#) na Referência de AWS CLI Comandos.

restore-managed-prefix-list-version

O código de exemplo a seguir mostra como usar `restore-managed-prefix-list-version`.

AWS CLI

US-West-2**Para restaurar uma versão da lista de prefixos**

O seguinte `restore-managed-prefix-list-version` restaura as entradas da versão 1 da lista de prefixos especificada.

```
aws ec2 restore-managed-prefix-list-version \  
  --prefix-list-id pl-0111222333444abcd
```



```
--prefix-list-id pl-0123456abcabc1 \  
--current-version 2 \  
--previous-version 1
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Listas de prefixos gerenciadas](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [RestoreManagedPrefixListVersion](#) na Referência de AWS CLI Comandos.

restore-snapshot-from-recycle-bin

O código de exemplo a seguir mostra como usar `restore-snapshot-from-recycle-bin`.

AWS CLI

Para restaurar instantâneos da Lixeira

O `restore-snapshot-from-recycle-bin` exemplo a seguir restaura um instantâneo da Lixeira. Quando você restaura um snapshot da lixeira, ele fica imediatamente disponível para uso e é removido da lixeira. É possível usar um snapshot restaurado como usa qualquer outro snapshot em sua conta.

```
aws ec2 restore-snapshot-from-recycle-bin \  
--snapshot-id snap-01234567890abcdef
```

Este comando não produz saída.

Para obter mais informações sobre a Lixeira para a AmazonEBS, consulte [Recuperar instantâneos da Lixeira no](#) Guia do usuário da Amazon EC2.

- Para API obter detalhes, consulte [RestoreSnapshotFromRecycleBin](#) na Referência de AWS CLI Comandos.

restore-snapshot-tier

O código de exemplo a seguir mostra como usar `restore-snapshot-tier`.

AWS CLI

Exemplo 1: Para restaurar permanentemente um instantâneo arquivado

O `restore-snapshot-tier` exemplo a seguir restaura permanentemente o instantâneo especificado. Especifique `--snapshot-id` e inclua a `permanent-restore` opção.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

Saída:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

Para obter mais informações sobre arquivamento de instantâneos, consulte [Arquivar EBS instantâneos da Amazon <https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/snapshot-archive.html>](https://docs.aws.amazon.com/AWS-EC2/latest/UserGuide/snapshot-archive.html) no Guia do usuário da Amazon. EC2

Exemplo 2: Para restaurar temporariamente um instantâneo arquivado

O `restore-snapshot-tier` exemplo a seguir restaura temporariamente o instantâneo especificado. Omita a opção `--permanent-restore`. Especifique `--snapshot-id` e, para `temporary-restore-days`, especifique o número de dias durante os quais restaurar o snapshot. `temporary-restore-days` deve ser especificado em dias. O intervalo permitido é 1 de 180. Se você não especificar um valor, o padrão de 1 dia será adotado.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 5
```

Saída:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 5,  
  "IsPermanentRestore": false  
}
```

Para obter mais informações sobre arquivamento de instantâneos, consulte Arquivar EBS instantâneos da Amazon <<https://docs.aws.amazon.com/AWS EC2/latest/ /snapshot-archive.html> UserGuide > no Guia do usuário da Amazon. EC2

Exemplo 3: Para modificar o período de restauração

O `restore-snapshot-tier` exemplo a seguir altera o período de restauração do snapshot especificado para 10 dias.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 10
```

Saída:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 10,  
  "IsPermanentRestore": false  
}
```

Para obter mais informações sobre arquivamento de instantâneos, consulte Arquivar EBS instantâneos da Amazon <<https://docs.aws.amazon.com/AWS EC2/latest/ /snapshot-archive.html> UserGuide > no Guia do usuário da Amazon. EC2

Exemplo 4: Para modificar o tipo de restauração

O `restore-snapshot-tier` exemplo a seguir altera o tipo de restauração do snapshot especificado de temporário para permanente.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

Saída:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

Para obter mais informações sobre arquivamento de instantâneos, consulte Arquivar EBS instantâneos da Amazon <https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html> no Guia do usuário da Amazon. EC2

- Para API obter detalhes, consulte [RestoreSnapshotTier](#) na Referência de AWS CLI Comandos.

revoke-client-vpn-ingress

O código de exemplo a seguir mostra como usar `revoke-client-vpn-ingress`.

AWS CLI

Para revogar uma regra de autorização para um endpoint do cliente VPN

O `revoke-client-vpn-ingress` exemplo a seguir revoga uma regra de acesso à Internet (`0.0.0.0/0`) para todos os grupos.

```
aws ec2 revoke-client-vpn-ingress \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

Saída:

```
{  
  "Status": {  
    "Code": "revoking"  
  }  
}
```

Para obter mais informações, consulte [Regras de autorização](#) no Guia do VPN administrador do AWS cliente.

- Para API obter detalhes, consulte [RevokeClientVpnIngress](#) na Referência de AWS CLI Comandos.

revoke-security-group-egress

O código de exemplo a seguir mostra como usar `revoke-security-group-egress`.

AWS CLI

Exemplo 1: Para remover a regra que permite tráfego de saída para um intervalo de endereços específico

O comando de `revoke-security-group-egress` exemplo a seguir remove a regra que concede acesso aos intervalos de endereços especificados na 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}]
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de segurança](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para remover a regra que permite tráfego de saída para um grupo de segurança específico

O comando de `revoke-security-group-egress` exemplo a seguir remove a regra que concede acesso ao grupo de segurança especificado na 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"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de segurança](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [RevokeSecurityGroupEgress](#) na Referência de AWS CLI Comandos.

revoke-security-group-ingress

O código de exemplo a seguir mostra como usar `revoke-security-group-ingress`.

AWS CLI

Exemplo 1: Para remover uma regra de um grupo de segurança

O `revoke-security-group-ingress` exemplo a seguir remove o acesso à TCP porta 22 para o intervalo de `203.0.113.0/24` endereços do grupo de segurança especificado como padrãoVPC.

```
aws ec2 revoke-security-group-ingress \  
  --group-name mySecurityGroup \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

Esse comando não produzirá nenhuma saída se for bem-sucedido.

Para obter mais informações, consulte [Grupos de segurança](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para remover uma regra usando o conjunto de permissões de IP

O `revoke-security-group-ingress` exemplo a seguir usa o `ip-permissions` parâmetro para remover uma regra de entrada que permite a ICMP mensagem Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Tipo 3, Código 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}]
```

Esse comando não produzirá nenhuma saída se for bem-sucedido.

Para obter mais informações, consulte [Grupos de segurança](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [RevokeSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

run-instances

O código de exemplo a seguir mostra como usar `run-instances`.

AWS CLI

Exemplo 1: executar uma instância em uma sub-rede padrão

O `run-instances` exemplo a seguir inicia uma única instância do tipo `t2.micro` na sub-rede padrão da região atual e a associa à sub-rede padrão VPC da região. O key pair é opcional se você não planeja se conectar à sua instância usando SSH (Linux) ou RDP (Windows).

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --key-name MyKeyPair
```

Saída:

```
{  
  "Instances": [  
    {  
      "AmiLaunchIndex": 0,  
      "ImageId": "ami-0abcdef1234567890",  
      "InstanceId": "i-1231231230abcdef0",  
      "InstanceType": "t2.micro",  
      "KeyName": "MyKeyPair",  
      "LaunchTime": "2018-05-10T08:05:20.000Z",  
      "Monitoring": {  
        "State": "disabled"  
      },  
      "Placement": {  
        "AvailabilityZone": "us-east-2a",  
        "GroupName": "",  
        "Tenancy": "default"  
      },  
      "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",  
      "PrivateIpAddress": "10.0.0.157",  
      "ProductCodes": [],
```

```
"PublicDnsName": "",
"State": {
  "Code": 0,
  "Name": "pending"
},
"StateTransitionReason": "",
"SubnetId": "subnet-04a636d18e83cfac",
"VpcId": "vpc-1234567890abcdef0",
"Architecture": "x86_64",
"BlockDeviceMappings": [],
"ClientToken": "",
"EbsOptimized": false,
"Hypervisor": "xen",
"NetworkInterfaces": [
  {
    "Attachment": {
      "AttachTime": "2018-05-10T08:05:20.000Z",
      "AttachmentId": "eni-attach-0e325c07e928a0405",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching"
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "MySecurityGroup",
        "GroupId": "sg-0598c7d356eba48d7"
      }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "0a:ab:58:e0:67:e2",
    "NetworkInterfaceId": "eni-0c0a29997760baee7",
    "OwnerId": "123456789012",
    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
    "PrivateIpAddress": "10.0.0.157",
    "PrivateIpAddresses": [
      {
        "Primary": true,
        "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
        "PrivateIpAddress": "10.0.0.157"
      }
    ],
    "SourceDestCheck": true,
```



```
        "Status": "in-use",
        "SubnetId": "subnet-04a636d18e83cfacb",
        "VpcId": "vpc-1234567890abcdef0",
        "InterfaceType": "interface"
    }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupName": "MySecurityGroup",
        "GroupId": "sg-0598c7d356eba48d7"
    }
],
"SourceDestCheck": true,
"StateReason": {
    "Code": "pending",
    "Message": "pending"
},
"Tags": [],
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 1
},
"CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
},
"MetadataOptions": {
    "State": "pending",
    "HttpTokens": "optional",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "enabled"
}
}
],
"OwnerId": "123456789012",
"ReservationId": "r-02a3f596d91211712"
}
```

Exemplo 2: para executar uma instância em uma sub-rede não padrão e adicionar um endereço IP público

O exemplo `run-instances` a seguir solicita um endereço IP público para uma instância que você está executando em uma sub-rede não padrão. A instância está associada ao grupo de segurança especificado.

```
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
```

Para obter um exemplo da saída de `run-instances`, consulte o Exemplo 1.

Exemplo 3: executar uma instância com volumes adicionais

O exemplo `run-instances` a seguir usa um mapeamento de dispositivos de blocos, especificado em `mapping.json`, para anexar volumes adicionais na execução. Um mapeamento de dispositivos de blocos pode especificar EBS volumes, volumes de armazenamento de instâncias ou EBS volumes e volumes de armazenamento de instâncias.

```
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
```

Conteúdo de `mapping.json`. Este exemplo adiciona `/dev/sdh` um EBS volume vazio com um tamanho de 100 GiB.

```
[  
  {  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
      "VolumeSize": 100  
    }  
  }  
]
```

Conteúdo de `mapping.json`. Este exemplo adiciona `ephemeral1` como um volume de armazenamento de instância.

```
[
  {
    "DeviceName": "/dev/sdc",
    "VirtualName": "ephemeral1"
  }
]
```

Para obter um exemplo da saída de `run-instances`, consulte o Exemplo 1.

Para obter mais informações sobre mapeamentos de dispositivos de blocos, consulte [Mapeamento de dispositivos de blocos](#) no Guia EC2 do usuário da Amazon.

Exemplo 4: executar uma instância e adicionar tags na criação

O exemplo `run-instances` a seguir adiciona uma tag com uma chave de `webserver` e um valor de `production` à instância. O comando também aplica uma tag com uma chave de `cost-center` e um valor de `cc123` a qualquer EBS volume criado (nesse caso, o volume raiz).

```
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}]'
```

Para obter um exemplo da saída de `run-instances`, consulte o Exemplo 1.

Exemplo 5: executar uma instância com dados do usuário

O exemplo `run-instances` a seguir passa dados do usuário em um arquivo chamado `my_script.txt` que contém um script de configuração para a sua instância. O script é executado na inicialização.

```
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
```

Para obter um exemplo da saída de `run-instances`, consulte o Exemplo 1.

Para obter mais informações sobre os dados do usuário da instância, consulte Como [trabalhar com dados do usuário da instância](#) no Guia EC2 do usuário da Amazon.

Exemplo 6: executar uma instância de desempenho expansível

O exemplo `run-instances` a seguir executa uma instância `t2.micro` com a opção de crédito `unlimited`. Ao executar uma instância T2, se você não especificar a `--credit-specification`, o padrão é a opção de crédito `standard`. Ao executar uma instância T3, o padrão é a opção de crédito `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
```

Para obter um exemplo da saída de `run-instances`, consulte o Exemplo 1.

Para obter mais informações sobre instâncias de desempenho intermitentes, consulte Instâncias de [desempenho intermitentes](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [RunInstances](#) na Referência de AWS CLI Comandos.

run-scheduled-instances

O código de exemplo a seguir mostra como usar `run-scheduled-instances`.

AWS CLI

Para iniciar uma instância programada

Este exemplo inicia a instância agendada especificada em umVPC.

Comando:

```
aws ec2 run-scheduled-instances --scheduled-instance-id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-specification file://launch-specification.json
```

Especificação de lançamento.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"
  }
}
```

Saída:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

Este exemplo inicia a instância agendada especificada em EC2 -Classic.

Comando:

```
aws ec2 run-scheduled-instances --scheduled-instance-id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-specification file://launch-specification.json
```

Especificação de lançamento.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
  "IamInstanceProfile": {
    "Name": "my-iam-role"
  }
}
```

Saída:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

- Para API obter detalhes, consulte [RunScheduledInstances](#) na Referência de AWS CLI Comandos.

search-local-gateway-routes

O código de exemplo a seguir mostra como usar `search-local-gateway-routes`.

AWS CLI

Para pesquisar rotas em uma tabela de rotas do gateway local

O `search-local-gateway-routes` exemplo a seguir pesquisa rotas estáticas na tabela de rotas do gateway local especificada.

```
aws ec2 search-local-gateway-routes \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
  --filters "Name=type,Values=static"
```

Saída:

```
{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
  }
}
```

- Para API obter detalhes, consulte [SearchLocalGatewayRoutes](#) na Referência de AWS CLI Comandos.

search-transit-gateway-multicast-groups

O código de exemplo a seguir mostra como usar `search-transit-gateway-multicast-groups`.

AWS CLI

Para pesquisar um ou mais grupos multicast do Transit Gateway e retornar as informações de associação ao grupo

O `search-transit-gateway-multicast-groups` exemplo a seguir retorna a associação do grupo multicast especificado.

```
aws ec2 search-transit-gateway-multicast-groups \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Gerenciando grupos multicast](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [SearchTransitGatewayMulticastGroups](#) na Referência de AWS CLI Comandos.

search-transit-gateway-routes

O código de exemplo a seguir mostra como usar `search-transit-gateway-routes`.

AWS CLI

Para pesquisar rotas na tabela de rotas do gateway de trânsito especificada

O `search-transit-gateway-routes` exemplo a seguir retorna todas as rotas do tipo `static` na tabela de rotas especificada.

```

aws ec2 search-transit-gateway-routes \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
  --filters "Name=type,Values=static"

```

Saída:

```

{
  "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
}
```

Para obter mais informações, consulte as [tabelas de rotas do Transit Gateway](#) no Transit Gateways Guide.

- Para API obter detalhes, consulte [SearchTransitGatewayRoutes](#) na Referência de AWS CLI Comandos.

send-diagnostic-interrupt

O código de exemplo a seguir mostra como usar `send-diagnostic-interrupt`.

AWS CLI

Para enviar uma interrupção de diagnóstico

O `send-diagnostic-interrupt` exemplo a seguir envia uma interrupção de diagnóstico para a instância especificada.

```
aws ec2 send-diagnostic-interrupt \  
  --instance-id i-1234567890abcdef0
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SendDiagnosticInterrupt](#) na Referência de AWS CLI Comandos.

start-instances

O código de exemplo a seguir mostra como usar `start-instances`.

AWS CLI

Para iniciar uma EC2 instância da Amazon

Este exemplo inicia a instância especificada com EBS suporte da Amazon.

Comando:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Saída:

```
{
  "StartingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}
```

Para obter mais informações, consulte Interromper e iniciar sua instância no Guia do usuário do Amazon Elastic Compute Cloud.

- Para API obter detalhes, consulte [StartInstances](#) na Referência de AWS CLI Comandos.

start-network-insights-access-scope-analysis

O código de exemplo a seguir mostra como usar `start-network-insights-access-scope-analysis`.

AWS CLI

Para iniciar uma análise do escopo de acesso do Network Insights

O `start-network-insights-access-scope-analysis` exemplo a seguir inicia a análise do escopo em sua AWS conta.

```
aws ec2 start-network-insights-access-scope-analysis \
  --region us-east-1 \
  --network-insights-access-scope-id nis-123456789111
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Introdução ao Network Access Analyzer usando o Guia do AWS CLI Network Access Analyzer](#).

- Para API obter detalhes, consulte [StartNetworkInsightsAccessScopeAnalysis](#) na Referência de AWS CLI Comandos.

start-network-insights-analysis

O código de exemplo a seguir mostra como usar `start-network-insights-analysis`.

AWS CLI

Para analisar um caminho

O `start-network-insights-analysis` exemplo a seguir analisa o caminho entre a origem e o destino. Para visualizar os resultados da análise do caminho, use o `describe-network-insights-analyses` comando.

```
aws ec2 start-network-insights-analysis \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Introdução ao uso do AWS CLI no Guia](#) do Reachability Analyzer.

- Para API obter detalhes, consulte [StartNetworkInsightsAnalysis](#) na Referência de AWS CLI Comandos.

start-vpc-endpoint-service-private-dns-verification

O código de exemplo a seguir mostra como usar `start-vpc-endpoint-service-private-dns-verification`.

AWS CLI

Para iniciar o processo de DNS verificação

O `start-vpc-endpoint-service-private-dns-verification` exemplo a seguir inicia o processo DNS de verificação para o serviço de endpoint especificado.

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \  
  --service-id vpce-svc-071afff70666e61e0
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar DNS nomes](#) no Guia AWS PrivateLink do usuário.

- Para API obter detalhes, consulte [StartVpcEndpointServicePrivateDnsVerification](#) na Referência de AWS CLI Comandos.

stop-instances

O código de exemplo a seguir mostra como usar stop-instances.

AWS CLI

Exemplo 1: Para parar uma EC2 instância da Amazon

O stop-instances exemplo a seguir interrompe a instância especificada EBS com suporte da Amazon.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0
```

Saída:

```
{  
  "StoppingInstances": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "PreviousState": {  
        "Code": 16,  
        "Name": "running"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Interromper e iniciar sua instância](#) no Guia do usuário do Amazon Elastic Compute Cloud.

Exemplo 2: Para hibernar uma instância da Amazon EC2

O `stop-instances` exemplo a seguir hiberna a instância EBS apoiada pela Amazon se a instância estiver habilitada para hibernação e atender aos pré-requisitos de hibernação. Depois de colocar a instância em hibernação, ela é interrompida.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0 \  
  --hibernate
```

Saída:

```
{  
  "StoppingInstances": [  
    {  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "InstanceId": "i-1234567890abcdef0",  
      "PreviousState": {  
        "Code": 16,  
        "Name": "running"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Colocar em hibernação uma instância sob demanda do Linux](#) no Guia do usuário do Amazon Elastic Cloud Compute.

- Para API obter detalhes, consulte [StopInstances](#) na Referência de AWS CLI Comandos.

terminate-client-vpn-connections

O código de exemplo a seguir mostra como usar `terminate-client-vpn-connections`.

AWS CLI

Para encerrar uma conexão com um endpoint do cliente VPN

O `terminate-client-vpn-connections` exemplo a seguir encerra a conexão especificada com o VPN endpoint do cliente.

```
aws ec2 terminate-client-vpn-connections \  
  --client-vpn-endpoint-id vpn-endpoint-123456789123abcde \  
  --connection-id cvpn-connection-04edd76f5201e0cb8
```

Saída:

```
{  
  "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",  
  "ConnectionStatuses": [  
    {  
      "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",  
      "PreviousStatus": {  
        "Code": "active"  
      },  
      "CurrentStatus": {  
        "Code": "terminating"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Conexões do AWS cliente](#) no Guia do VPN administrador do cliente.

- Para API obter detalhes, consulte [TerminateClientVpnConnections](#) na Referência de AWS CLI Comandos.

terminate-instances

O código de exemplo a seguir mostra como usar terminate-instances.

AWS CLI

Para encerrar uma instância da Amazon EC2

Este exemplo encerra a instância especificada.

Comando:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

Saída:

```
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

Para obter mais informações, consulte Como usar EC2 instâncias da Amazon no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [TerminateInstances](#) na Referência de AWS CLI Comandos.

unassign-ipv6-addresses

O código de exemplo a seguir mostra como usar unassign-ipv6-addresses.

AWS CLI

Para cancelar a atribuição de um IPv6 endereço de uma interface de rede

Este exemplo cancela a atribuição do IPv6 endereço especificado da interface de rede especificada.

Comando:

```
aws ec2 unassign-ipv6-addresses --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

Saída:

```
{
  "NetworkInterfaceId": "eni-23c49b68",
```



```
"UnassignedIpv6Addresses": [  
  "2001:db8:1234:1a00:3304:8879:34cf:4071"  
]  
}
```

- Para API obter detalhes, consulte [UnassignIpv6Addresses](#) in AWS CLI Command Reference.

unassign-private-ip-addresses

O código de exemplo a seguir mostra como usar `unassign-private-ip-addresses`.

AWS CLI

Para cancelar a atribuição de um endereço IP privado secundário de uma interface de rede

Este exemplo cancela a atribuição do endereço IP privado especificado da interface de rede especificada. Se o comando for bem-sucedido, nenhuma saída será retornada.

Comando:

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-  
ip-addresses 10.0.0.82
```

- Para API obter detalhes, consulte [UnassignPrivateIpAddresses](#) na Referência de AWS CLI Comandos.

unassign-private-nat-gateway-address

O código de exemplo a seguir mostra como usar `unassign-private-nat-gateway-address`.

AWS CLI

Para cancelar a atribuição de um endereço IP privado do seu gateway privado NAT

O `unassign-private-nat-gateway-address` exemplo a seguir cancela a atribuição do endereço IP especificado do gateway privado especificado. NAT

```
aws ec2 unassign-private-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --private-ip-addresses 10.0.20.197
```

Saída:

```
{
  "NatGatewayId": "nat-0ee3edd182361f662",
  "NatGatewayAddresses": [
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "PrivateIp": "10.0.20.197",
      "IsPrimary": false,
      "Status": "unassigning"
    }
  ]
}
```

Para obter mais informações, consulte [NATgateways](#) no Guia do VPC usuário da Amazon.

- Para API obter detalhes, consulte [UnassignPrivateNatGatewayAddress](#) na Referência de AWS CLI Comandos.

unlock-snapshot

O código de exemplo a seguir mostra como usar `unlock-snapshot`.

AWS CLI

Para desbloquear um instantâneo

O `unlock-snapshot` exemplo a seguir desbloqueia o instantâneo especificado.

```
aws ec2 unlock-snapshot \
  --snapshot-id snap-0b5e733b4a8df6e0d
```

Saída:

```
{
  "SnapshotId": "snap-0b5e733b4a8df6e0d"
}
```

Para obter mais informações, consulte [Snapshot lock](#) no Guia do EBS usuário da Amazon.

- Para API obter detalhes, consulte [UnlockSnapshot](#) na Referência de AWS CLI Comandos.

unmonitor-instances

O código de exemplo a seguir mostra como usar `unmonitor-instances`.

AWS CLI

Para desabilitar o monitoramento detalhado de uma instância

Este exemplo de comando desabilita o monitoramento detalhado da instância especificada.

Comando:

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

Saída:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "disabling"
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [UnmonitorInstances](#) na Referência de AWS CLI Comandos.

update-security-group-rule-descriptions-egress

O código de exemplo a seguir mostra como usar `update-security-group-rule-descriptions-egress`.

AWS CLI

Para atualizar a descrição de uma regra de grupo de segurança de saída

O `update-security-group-rule-descriptions-egress` exemplo a seguir atualiza a descrição da regra do grupo de segurança para a porta e o intervalo de IPv4 endereços especificados. A descrição 'Outbound HTTP access to server 2' substitui qualquer descrição existente para a regra.

```
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"}]
```

Saída:

```
{  
  "Return": true  
}
```

Para obter mais informações, consulte [Regras de grupos de segurança](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateSecurityGroupRuleDescriptionsEgress](#) na Referência de AWS CLI Comandos.

update-security-group-rule-descriptions-ingress

O código de exemplo a seguir mostra como usar `update-security-group-rule-descriptions-ingress`.

AWS CLI

Exemplo 1: Para atualizar a descrição de uma regra de grupo de segurança de entrada com uma fonte CIDR

O `update-security-group-rule-descriptions-ingress` exemplo a seguir atualiza a descrição da regra do grupo de segurança para a porta e o intervalo de IPv4 endereços especificados. A descrição 'SSH access from ABC office' substitui qualquer descrição existente para a regra.

```
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"}]'
```

Saída:

```
{
  "Return": true
}
```

Para obter mais informações, consulte [Regras de grupos de segurança](#) no Guia EC2 do usuário da Amazon.

Exemplo 2: Para atualizar a descrição de uma regra de grupo de segurança de entrada com uma fonte de lista de prefixos

O `update-security-group-rule-descriptions-ingress` exemplo a seguir atualiza a descrição da regra do grupo de segurança para a porta especificada e a lista de prefixos. A descrição 'SSH access from ABC office' substitui qualquer descrição existente para a regra.

```
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"}]'
```

Saída:

```
{
  "Return": true
}
```

Para obter mais informações, consulte [Regras de grupos de segurança](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateSecurityGroupRuleDescriptionsIngress](#) na Referência de AWS CLI Comandos.

withdraw-byoip-cidr

O código de exemplo a seguir mostra como usar `withdraw-byoip-cidr`.

AWS CLI

Para parar de anunciar um intervalo de endereços

O `withdraw-byoip-cidr` exemplo a seguir interrompe a publicidade do intervalo de endereços especificado.

```
aws ec2 withdraw-byoip-cidr
  --cidr 203.0.113.25/24
```

Saída:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    "State": "advertised"
  }
}
```

- Para API obter detalhes, consulte [WithdrawByoipCidr](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon EC2 Instance Connect usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon EC2 Instance Connect.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

send-ssh-public-key

O código de exemplo a seguir mostra como usar `send-ssh-public-key`.

AWS CLI

Para enviar uma chave SSH pública para uma instância

O `send-ssh-public-key` exemplo a seguir envia a chave SSH pública especificada para a instância especificada. A chave é usada para autenticar o usuário especificado.

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SendSshPublicKey](#) na Referência de AWS CLI Comandos.

ECRExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon ECR.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-check-layer-availability

O código de exemplo a seguir mostra como usar `batch-check-layer-availability`.

AWS CLI

Para verificar a disponibilidade de uma camada

O `batch-check-layer-availability` exemplo a seguir verifica a disponibilidade de uma camada com o resumo `sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed` no `cluster-autoscaler` repositório.

```
aws ecr batch-check-layer-availability \  
  --repository-name cluster-autoscaler \  
  --layer-  
  digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

Saída:

```
{  
  "layers": [  
    {  
      "layerDigest":  
      "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",  
      "layerAvailability": "AVAILABLE",  
      "layerSize": 2777,  
      "mediaType": "application/vnd.docker.container.image.v1+json"  
    }  
  ],  
  "failures": []  
}
```

- Para API obter detalhes, consulte [BatchCheckLayerAvailability](#) na Referência de AWS CLI Comandos.

batch-delete-image

O código de exemplo a seguir mostra como usar `batch-delete-image`.

AWS CLI

Exemplo 1: Para excluir uma imagem

O `batch-delete-image` exemplo a seguir exclui uma imagem com a tag `precise` no repositório especificado no registro padrão de uma conta.

```
aws ecr batch-delete-image \  
  --repository-name ubuntu \  
  --tag-name precise
```



```
--image-ids imageTag=precise
```

Saída:

```
{
  "failures": [],
  "imageIds": [
    {
      "imageTag": "precise",
      "imageDigest":
"sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"
    }
  ]
}
```

Exemplo 2: Para excluir várias imagens

O `batch-delete-image` exemplo a seguir exclui todas as imagens marcadas com `prod` e `team1` no repositório especificado.

```
aws ecr batch-delete-image \
  --repository-name MyRepository \
  --image-ids imageTag=prod imageTag=team1
```

Saída:

```
{
  "imageIds": [
    {
      "imageDigest": "sha256:123456789012",
      "imageTag": "prod"
    },
    {
      "imageDigest": "sha256:567890121234",
      "imageTag": "team1"
    }
  ],
  "failures": []
}
```

Para obter mais informações, consulte [Excluindo uma imagem](#) no Guia do ECR usuário da Amazon.

- Para API obter detalhes, consulte [BatchDeletelma](#) na Referência de AWS CLI Comandos.

batch-get-image

O código de exemplo a seguir mostra como usar `batch-get-image`.

AWS CLI

Exemplo 1: Para obter uma imagem

O `batch-get-image` exemplo a seguir obtém uma imagem com a tag `v1.13.6` em um repositório chamado `cluster-autoscaler` no registro padrão de uma conta.

```
aws ecr batch-get-image \
  --repository-name cluster-autoscaler \
  --image-ids imageTag=v1.13.6
```

Saída:

```
{
  "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  \"layers\": [\n    {\n      \"mediaType
  \": \"application/vnd.docker.image.rootfs.diff.tar.gzip
  \", \n      \"size\": 17743696, \n      \"digest\":
  \"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffef5a85f900caaf6e1\"\n
  }, \n    {\n      \"mediaType\": \"application/
  vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 2565026, \n
  \"digest\":
  \"sha256:8c8a779d3a537b767ae1091fe6e00c2590afd16767aa6096d1b318d75494819f
```

```

{"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip", "size": 28005981, "digest": "sha256:c44ba47496991c9982ee493b47fd25c252caabf2b4ae7dd679c9a27b6a3c8fb7"}, {"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip", "size": 775, "digest": "sha256:e2c388b44226544363ca007be7b896bcce1baebea04da23cbd165eac30be650f"}]
  ],
  "failures": []
}

```

Exemplo 2: Para obter várias imagens

O `batch-get-image` exemplo a seguir exibe detalhes de todas as imagens marcadas com `prod` e `team1` no repositório especificado.

```

aws ecr batch-get-image \
  --repository-name MyRepository \
  --image-ids imageTag=prod imageTag=team1

```

Saída:

```

{
  "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": []
}
```

Para obter mais informações, consulte [Imagens](#) no Guia do ECR usuário da Amazon.

- Para API obter detalhes, consulte [BatchGetImage](#) na Referência de AWS CLI Comandos.

complete-layer-upload

O código de exemplo a seguir mostra como usar `complete-layer-upload`.

AWS CLI

Para concluir o upload de uma camada de imagem

O `complete-layer-upload` exemplo a seguir conclui o upload de uma camada de imagem para o `layer-test` repositório.

```
aws ecr complete-layer-upload \
  --repository-name layer-test \
  --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
  --layer-digests 6cb64b8a-9378-0e33-2ab1-
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

Saída:

```
{
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
  "layerDigest":
  "sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",
  "repositoryName": "layer-test",
  "registryId": "130757420319"
}
```

- Para API obter detalhes, consulte [CompleteLayerUpload](#) na Referência de AWS CLI Comandos.

create-repository

O código de exemplo a seguir mostra como usar `create-repository`.

AWS CLI

Exemplo 1: Para criar um repositório

O `create-repository` exemplo a seguir cria um repositório dentro do namespace especificado no registro padrão de uma conta.

```
aws ecr create-repository \  
  --repository-name project-a/nginx-web-app
```

Saída:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/  
nginx-web-app"  
  }  
}
```

Para obter mais informações, consulte [Criação de um repositório](#) no Guia do ECR usuário da Amazon.

Exemplo 2: Para criar um repositório configurado com imutabilidade da tag de imagem

O `create-repository` exemplo a seguir cria um repositório configurado para imutabilidade de tags no registro padrão de uma conta.

```
aws ecr create-repository \  
  --repository-name sample-repo \  
  --image-tag-mutability IMMUTABLE
```

Saída:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/sample-  
repo",
```

```
    "imageTagMutability": "IMMUTABLE"
  }
}
```

Para obter mais informações, consulte [Image Tag Mutability](#) no Guia do ECR usuário da Amazon.

Exemplo 3: Para criar um repositório configurado com uma configuração de digitalização

O `create-repository` exemplo a seguir cria um repositório configurado para realizar uma verificação de vulnerabilidade no envio de imagens no registro padrão de uma conta.

```
aws ecr create-repository \
  --repository-name sample-repo \
  --image-scanning-configuration scanOnPush=true
```

Saída:

```
{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "sample-repo",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/sample-
repo",
    "imageScanningConfiguration": {
      "scanOnPush": true
    }
  }
}
```

Para obter mais informações, consulte [Digitalização de imagens](#) no Guia ECR do usuário da Amazon.

- Para API obter detalhes, consulte [CreateRepository](#) na Referência de AWS CLI Comandos.

delete-lifecycle-policy

O código de exemplo a seguir mostra como usar `delete-lifecycle-policy`.

AWS CLI

Para excluir a política de ciclo de vida de um repositório

O `delete-lifecycle-policy` exemplo a seguir exclui a política de ciclo de vida do repositório `hello-world`

```
aws ecr delete-lifecycle-policy \  
  --repository-name hello-world
```

Saída:

```
{  
  "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  
}
```

- Para API obter detalhes, consulte [DeleteLifecyclePolicy](#) na Referência de AWS CLI Comandos.

`delete-repository-policy`

O código de exemplo a seguir mostra como usar `delete-repository-policy`.

AWS CLI

Para excluir a política de repositório de um repositório

O `delete-repository-policy` exemplo a seguir exclui a política de repositório do `cluster-autoscaler` repositório.

```
aws ecr delete-repository-policy \  
  --repository-name cluster-autoscaler
```

Saída:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "cluster-autoscaler",  
  "policyText": "{\"Version\" : \"2008-10-17\", \"Statement\" : [ {  
\"Sid\" : \"allow public pull\", \"Effect\" : \"Allow\", \"Principal\" :
```

```

\)*",\n  \Action" : [ \ecr:BatchCheckLayerAvailability", \ecr:BatchGetImage
", \ecr:GetDownloadUrlForLayer" ]\n } ]\n}"
}

```

- Para API obter detalhes, consulte [DeleteRepositoryPolicy](#) na Referência de AWS CLI Comandos.

delete-repository

O código de exemplo a seguir mostra como usar `delete-repository`.

AWS CLI

Para excluir um repositório

O `delete-repository` exemplo de force de comando a seguir exclui o repositório especificado no registro padrão de uma conta. O `--force` sinalizador é obrigatório se o repositório contiver imagens.

```

aws ecr delete-repository \
  --repository-name ubuntu \
  --force

```

Saída:

```

{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "ubuntu",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"
  }
}

```

Para obter mais informações, consulte [Excluindo um repositório no Guia ECR](#) do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteRepository](#) na Referência de AWS CLI Comandos.

describe-image-scan-findings

O código de exemplo a seguir mostra como usar `describe-image-scan-findings`.

AWS CLI

Para descrever os resultados da digitalização de uma imagem

O `describe-image-scan-findings` exemplo a seguir retorna os resultados da digitalização de uma imagem usando o resumo da imagem no repositório especificado no registro padrão de uma conta.

```
aws ecr describe-image-scan-findings \
  --repository-name sample-repo \
  --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

Saída:

```
{
  "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."
}
}
```

Para obter mais informações, consulte [Digitalização de imagens](#) no Guia ECR do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeImageScanFindings](#) na Referência de AWS CLI Comandos.

describe-images

O código de exemplo a seguir mostra como usar `describe-images`.

AWS CLI

Para descrever uma imagem em um repositório

O `describe-images` exemplo a seguir exibe detalhes sobre uma imagem no `cluster-autoscaler` repositório com a tag. `v1.13.6`

```
aws ecr describe-images \
  --repository-name cluster-autoscaler \
  --image-ids imageTag=v1.13.6
```

Saída:

```
{
  "imageDetails": [
    {
      "registryId": "012345678910",
      "repositoryName": "cluster-autoscaler",
      "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
      "imageTags": [
        "v1.13.6"
      ],
      "imageSizeInBytes": 48318255,
      "imagePushedAt": 1565128275.0
    }
  ]
}
```

- Para API obter detalhes, consulte [Descrever Imagens](#) na Referência de AWS CLI Comandos.

describe-repositories

O código de exemplo a seguir mostra como usar `describe-repositories`.

AWS CLI

Como descrever os repositórios em um registro

Este exemplo descreve os repositórios no registro padrão de uma conta.

Comando:

```
aws ecr describe-repositories
```

Saída:

```
{
  "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"
    }
]
}
```

- Para API obter detalhes, consulte [DescribeRepositories](#) na Referência de AWS CLI Comandos.

get-authorization-token

O código de exemplo a seguir mostra como usar `get-authorization-token`.

AWS CLI

Para obter um token de autorização para seu registro padrão

O comando de `get-authorization-token` exemplo a seguir obtém um token de autorização para seu registro padrão.

```
aws ecr get-authorization-token
```

Saída:

```
{
  "authorizationData": [
    {
      "authorizationToken": "QVdT0kN...",
      "expiresAt": 1448875853.241,
      "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetAuthorizationToken](#) na Referência de AWS CLI Comandos.

get-download-url-for-layer

O código de exemplo a seguir mostra como usar `get-download-url-for-layer`.

AWS CLI

Para obter o download URL de uma camada

O `get-download-url-for-layer` exemplo a seguir exibe o download URL de uma camada com o resumo

`sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed`
no `cluster-autoscaler` repositório.

```
aws ecr get-download-url-for-layer \  
  --repository-name cluster-autoscaler \  
  --layer-  
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

Saída:

```
{  
  "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"  
}
```

- Para API obter detalhes, consulte [GetDownloadUrlForLayer](#) na Referência de AWS CLI Comandos.

`get-lifecycle-policy-preview`

O código de exemplo a seguir mostra como usar `get-lifecycle-policy-preview`.

AWS CLI

Para recuperar detalhes de uma prévia da política de ciclo de vida

O `get-lifecycle-policy-preview` exemplo a seguir recupera o resultado de uma prévia da política de ciclo de vida para o repositório especificado no registro padrão de uma conta.

Comando:

```
aws ecr get-lifecycle-policy-preview \
  --repository-name "project-a/amazon-ecs-sample"
```

Saída:

```
{
  "registryId": "012345678910",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\": 1,\n      \"description\": \"Expire images older than 14 days\",\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",\n        \"countNumber\": 14\n      },\n      \"action\": {\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}",
  "status": "COMPLETE",
  "previewResults": [],
  "summary": {
    "expiringImageTotalCount": 0
  }
}
```

Para obter mais informações, consulte [Políticas de ciclo de vida no Guia do ECR](#) usuário da Amazon.

- Para API obter detalhes, consulte [GetLifecyclePolicyPreview](#) na Referência de AWS CLI Comandos.

get-lifecycle-policy

O código de exemplo a seguir mostra como usar `get-lifecycle-policy`.

AWS CLI

Para recuperar uma política de ciclo de vida

O `get-lifecycle-policy` exemplo a seguir exibe detalhes da política de ciclo de vida do repositório especificado no registro padrão da conta.

```
aws ecr get-lifecycle-policy \
```

```
--repository-name "project-a/amazon-ecs-sample"
```

Saída:

```
{
  "registryId": "123456789012",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\"rules\": [{\"rulePriority\": 1, \"description\": \"Expire images older than 14 days\", \"selection\": {\"tagStatus\": \"untagged\", \"countType\": \"sinceImagePushed\", \"countUnit\": \"days\", \"countNumber\": 14}, \"action\": {\"type\": \"expire\"}}]}",
  "lastEvaluatedAt": 1504295007.0
}
```

Para obter mais informações, consulte [Políticas de ciclo de vida no Guia do ECR](#) usuário da Amazon.

- Para API obter detalhes, consulte [GetLifecyclePolicy](#) na Referência de AWS CLI Comandos.

get-login-password

O código de exemplo a seguir mostra como usar `get-login-password`.

AWS CLI

Para recuperar uma senha para se autenticar em um registro

A seguir, é `get-login-password` exibida uma senha que você pode usar com um cliente de contêiner de sua escolha para se autenticar em qualquer ECR registro da Amazon ao qual seu IAM diretor tenha acesso.

```
aws ecr get-login-password
```

Saída:

```
<password>
```

Para usar com o `DockerCLI`, canalize a saída do `get-login-password` comando para o `docker login` comando. Ao recuperar a senha, certifique-se de especificar a mesma região em que seu ECR registro da Amazon existe.

```
aws ecr get-login-password \
  --region <region> \
  | docker login \
  --username AWS \
  --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Para obter mais informações, consulte [Autenticação de registro](#) no Guia ECR do usuário da Amazon.

- Para API obter detalhes, consulte [GetLoginPassword](#) na Referência de AWS CLI Comandos.

get-login

O código de exemplo a seguir mostra como usar `get-login`.

AWS CLI

Para recuperar um comando de login do Docker para seu registro padrão

Este exemplo imprime um comando que você pode usar para fazer login no seu ECR registro padrão da Amazon.

Comando:

```
aws ecr get-login
```

Saída:

```
docker login -u AWS -p <password> -e none https://
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Para fazer login no registro de outra conta

Este exemplo imprime um ou mais comandos que você pode usar para fazer login nos ECR registros da Amazon associados a outras contas.

Comando:

```
aws ecr get-login --registry-ids 012345678910 023456789012
```


Saída:

```
docker login -u <username> -p <token-1> -e none <endpoint-1>
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- Para API obter detalhes, consulte [GetLogin](#) na Referência de AWS CLI Comandos.

get-repository-policy

O código de exemplo a seguir mostra como usar `get-repository-policy`.

AWS CLI

Para recuperar a política de repositório de um repositório

O `get-repository-policy` exemplo a seguir exibe detalhes sobre a política de repositório para o `cluster-autoscaler` repositório.

```
aws ecr get-repository-policy \
  --repository-name cluster-autoscaler
```

Saída:

```
{
  "registryId": "012345678910",
  "repositoryName": "cluster-autoscaler",
  "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :\n    \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\",\n    \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
```

- Para API obter detalhes, consulte [GetRepositoryPolicy](#) na Referência de AWS CLI Comandos.

initiate-layer-upload

O código de exemplo a seguir mostra como usar `initiate-layer-upload`.

AWS CLI

Para iniciar o upload de uma camada de imagem

O `initiate-layer-upload` exemplo a seguir inicia o upload de uma camada de imagem para o `layer-test` repositório.

```
aws ecr initiate-layer-upload \  
  --repository-name layer-test
```

Saída:

```
{  
  "partSize": 10485760,  
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"  
}
```

- Para API obter detalhes, consulte [InitiateLayerUpload](#) na Referência de AWS CLI Comandos.

list-images

O código de exemplo a seguir mostra como usar `list-images`.

AWS CLI

Como listar as imagens em um repositório

O exemplo de `list-images` a seguir exibe uma lista das imagens presentes no repositório `cluster-autoscaler`.

```
aws ecr list-images \  
  --repository-name cluster-autoscaler
```

Saída:

```
{  
  "imageIds": [  
    {  
      "imageDigest":  
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",  
      "imageTag": "v1.13.8"  
    },  
    {  
      "imageDigest":  
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
```

```

        "imageTag": "v1.13.7"
      },
      {
        "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
        "imageTag": "v1.13.6"
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ListImages](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags para o repositório

O `list-tags-for-resource` exemplo a seguir exibe uma lista das tags associadas ao `hello-world` repositório.

```

aws ecr list-tags-for-resource \
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world

```

Saída:

```

{
  "tags": [
    {
      "Key": "Stage",
      "Value": "Integ"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-image-scanning-configuration

O código de exemplo a seguir mostra como usar `put-image-scanning-configuration`.

AWS CLI

Para atualizar a configuração de digitalização de imagens para um repositório

O `put-image-scanning-configuration` exemplo a seguir atualiza a configuração de digitalização de imagens para o repositório especificado.

```
aws ecr put-image-scanning-configuration \  
  --repository-name sample-repo \  
  --image-scanning-configuration scanOnPush=true
```

Saída:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",  
  "imageScanningConfiguration": {  
    "scanOnPush": true  
  }  
}
```

Para obter mais informações, consulte [Digitalização de imagens](#) no Guia ECR do usuário da Amazon.

- Para API obter detalhes, consulte [PutImageScanningConfiguration](#) na Referência de AWS CLI Comandos.

`put-image-tag-mutability`

O código de exemplo a seguir mostra como usar `put-image-tag-mutability`.

AWS CLI

Para atualizar a configuração de mutabilidade da tag de imagem para um repositório

O `put-image-tag-mutability` exemplo a seguir configura o repositório especificado para imutabilidade de tags. Isso evita que todas as tags de imagem no repositório sejam sobrescritas.

```
aws ecr put-image-tag-mutability \  
  --repository-name hello-repository \  
  --image-tag-mutability IMMUTABLE
```

Saída:

```
{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageTagMutability": "IMMUTABLE"
}
```

Para obter mais informações, consulte [Image Tag Mutability](#) no Guia do ECR usuário da Amazon.

- Para API obter detalhes, consulte [PutImageTagMutability](#) na Referência de AWS CLI Comandos.

put-image

O código de exemplo a seguir mostra como usar `put-image`.

AWS CLI

Para remarcar uma imagem com seu manifesto

O `put-image` exemplo a seguir cria uma nova tag no `hello-world` repositório com um manifesto de imagem existente.

```
aws ecr put-image \
  --repository-name hello-world \
  --image-tag 2019.08 \
  --image-manifest file://hello-world.manifest.json
```

Conteúdo de `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"
    }
  ]
}

```

Saída:

```

{
  "image": {
    "registryId": "130757420319",
    "repositoryName": "hello-world",
    "imageId": {
      "imageDigest":
"sha256:8ece96b74f87652876199d83bd107d0435a196133af383ac54cb82b6cc5283ae",
      "imageTag": "2019.08"
    },
    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType
\": \"application/vnd.docker.distribution.manifest.v2+json
\",,\n  \"config\": {\n    \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",,\n    \"size\": 5695,\n    \"digest\":
\"sha256:cea5fe7701b7db3dd1c372f3cea6f43cdda444fcc488f530829145e426d8b980\"\n
  },\n  \"layers\": [\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 39096921,\n      \"digest
\": \"sha256:d8868e50ac4c7104d2200d42f432b661b2da8c1e417ccfae217e6a1e04bb9295\"\n

```

```

    },\n    {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 57938, \n        \"digest
\": \"sha256:83251ac64627fc331584f6c498b3aba5badc01574e2c70b2499af3af16630eed
\\\" \n    },\n    {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 423, \n        \"digest\":
\"sha256:589bba2f1b36ae56f0152c246e2541c5aa604b058febfcf2be32e9a304fec610\" \n    },
\n    {\n        \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n        \"size\": 680, \n        \"digest\":
\"sha256:d62ecaceda3964b735cdd2af613d6bb136a52c1da0838b2ff4b4dab4212bcb1c
\\\" \n    },\n    {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 162, \n        \"digest
\": \"sha256:6d93b41cfc6bf0d2522b7cf61588de4cd045065b36c52bd3aec2ba0622b2b22b
\\\" \n    },\n    {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 28268840, \n        \"digest
\": \"sha256:6986b4d4c07932c680b3587f2eac8b0e013568c003cc23b04044628a5c5e599f
\\\" \n    },\n    {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 35369152, \n        \"digest
\": \"sha256:8c5ec60f10102dc8da0649d866c7c2f706e459d0bdc25c83ad2de86f4996c276\" \n
    },\n    {\n        \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\", \n        \"size\": 155, \n        \"digest\":
\"sha256:cde50b1c594539c5f67cbede9aef95c9ae321ccfb857f7b251b45b84198adc85\" \n    },
\n    {\n        \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n        \"size\": 28737, \n        \"digest\":
\"sha256:2e102807ab72a73fc9abf53e8c50e421bdc337a0a8afcb242176edeec65977e4\" \n    },
\n    {\n        \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n        \"size\": 190, \n        \"digest\":
\"sha256:fc379bbd5ed37808772bef016553a297356c59b8f134659e6ee4ecb563c2f5a7\" \n    },
\n    {\n        \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip\",
\n        \"size\": 28748, \n        \"digest\":
\"sha256:021db240dfccf5a1aff19507d17c0177e5888e518acf295b52204b1825e8b7ee\" \n
    } \n ] \n }
}

```

- Para API obter detalhes, consulte [PutImagem](#) na Referência de AWS CLI Comandos.

put-lifecycle-policy

O código de exemplo a seguir mostra como usar `put-lifecycle-policy`.

AWS CLI

Para criar uma política de ciclo de vida

O `put-lifecycle-policy` exemplo a seguir cria uma política de ciclo de vida para o repositório especificado no registro padrão de uma conta.

```
aws ecr put-lifecycle-policy \
  --repository-name "project-a/amazon-ecs-sample" \
  --lifecycle-policy-text "file://policy.json"
```

Conteúdo de `policy.json`:

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "untagged",
        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 14
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

Saída:

```
{
  "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\"}}]}"
}
```

Para obter mais informações, consulte [Políticas de ciclo de vida no Guia do ECR](#) usuário da Amazon.

- Para API obter detalhes, consulte [PutLifecyclePolicy](#) na Referência de AWS CLI Comandos.

set-repository-policy

O código de exemplo a seguir mostra como usar `set-repository-policy`.

AWS CLI

Para definir a política de repositório para um repositório

O `set-repository-policy` exemplo a seguir anexa uma política de repositório contida em um arquivo ao `cluster-autoscaler` repositório.

```
aws ecr set-repository-policy \  
  --repository-name cluster-autoscaler \  
  --policy-text file://my-policy.json
```

Conteúdo de `my-policy.json`:

```
{  
  "Version" : "2008-10-17",  
  "Statement" : [  
    {  
      "Sid" : "allow public pull",  
      "Effect" : "Allow",  
      "Principal" : "*",  
      "Action" : [  
        "ecr:BatchCheckLayerAvailability",  
        "ecr:BatchGetImage",  
        "ecr:GetDownloadUrlForLayer"  
      ]  
    }  
  ]  
}
```

Saída:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "cluster-autoscaler",  
  "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :\n    \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\n    \", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
```

```
}
```

- Para API obter detalhes, consulte [SetRepositoryPolicy](#) na Referência de AWS CLI Comandos.

start-image-scan

O código de exemplo a seguir mostra como usar `start-image-scan`.

AWS CLI

Para iniciar uma verificação de vulnerabilidade de imagem

O `start-image-scan` exemplo a seguir inicia uma digitalização de imagem para e especificada pelo resumo da imagem no repositório especificado.

```
aws ecr start-image-scan \  
  --repository-name sample-repo \  
  --image-  
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

Saída:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",  
  "imageId": {  
    "imageDigest":  
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"  
  },  
  "imageScanStatus": {  
    "status": "IN_PROGRESS"  
  }  
}
```

Para obter mais informações, consulte [Digitalização de imagens](#) no Guia ECR do usuário da Amazon.

- Para API obter detalhes, consulte [StartImageScan](#) na Referência de AWS CLI Comandos.

start-lifecycle-policy-preview

O código de exemplo a seguir mostra como usar `start-lifecycle-policy-preview`.

AWS CLI

Para criar uma prévia da política de ciclo de vida

O `start-lifecycle-policy-preview` exemplo a seguir cria uma visualização prévia da política de ciclo de vida definida por um JSON arquivo para o repositório especificado.

```
aws ecr start-lifecycle-policy-preview \  
  --repository-name "project-a/amazon-ecs-sample" \  
  --lifecycle-policy-text "file://policy.json"
```

Conteúdo de `policy.json`:

```
{  
  "rules": [  
    {  
      "rulePriority": 1,  
      "description": "Expire images older than 14 days",  
      "selection": {  
        "tagStatus": "untagged",  
        "countType": "sinceImagePushed",  
        "countUnit": "days",  
        "countNumber": 14  
      },  
      "action": {  
        "type": "expire"  
      }  
    }  
  ]  
}
```

Saída:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "project-a/amazon-ecs-sample",  
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\": 1,\n      \"description\": \"Expire images older than 14\n      days\",\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",\n        \"countNumber\": 14\n      },\n      \"action\": {\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}"
```

```
"status": "IN_PROGRESS"
}
```

- Para API obter detalhes, consulte [StartLifecyclePolicyPreview](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um repositório

O `tag-resource` exemplo a seguir define uma tag com chave `Stage` e valor `Integ` no `hello-world` repositório.

```
aws ecr tag-resource \
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
  --tags Key=Stage,Value=Integ
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para desmarcar um repositório

O `untag-resource` exemplo a seguir remove a tag com a chave `Stage` do `hello-world` repositório.

```
aws ecr untag-resource \
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
  --tag-keys Stage
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

upload-layer-part

O código de exemplo a seguir mostra como usar `upload-layer-part`.

AWS CLI

Para carregar uma parte da camada

O seguinte `upload-layer-part` carrega uma parte da camada de imagem no `layer-test` repositório.

```
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
```

Saída:

```
{  
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",  
  "registryId": "012345678910",  
  "lastByteReceived": 8323314,  
  "repositoryName": "layer-test"  
}
```

- Para API obter detalhes, consulte [UploadLayerPart](#) na Referência de AWS CLI Comandos.

ECSExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonECS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-capacity-provider

O código de exemplo a seguir mostra como usar `create-capacity-provider`.

AWS CLI

Para criar um provedor de capacidade

O `create-capacity-provider` exemplo a seguir cria um provedor de capacidade que usa um grupo de Auto Scaling chamado MyASG, com escalabilidade gerenciada e proteção de terminação gerenciada ativada. Essa configuração é usada para o ECS escalonamento automático de clusters da 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"
```

Saída:

```
{
  "capacityProvider": {
    "capacityProviderArn": "arn:aws:ecs:us-east-1:123456789012:capacity-provider/
MyCapacityProvider",
    "name": "MyCapacityProvider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
      "autoScalingGroupArn": "arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/MyASG",
      "managedScaling": {
```

```
        "status": "ENABLED",
        "targetCapacity": 100,
        "minimumScalingStepSize": 1,
        "maximumScalingStepSize": 10000,
        "instanceWarmupPeriod": 300
    },
    "managedTerminationProtection": "ENABLED"
},
"tags": []
}
```

Para obter mais informações, consulte [Amazon ECS cluster auto scaling](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [CreateCapacityProvider](#) na Referência de AWS CLI Comandos.

create-cluster

O código de exemplo a seguir mostra como usar `create-cluster`.

AWS CLI

Exemplo 1: como criar um cluster

O exemplo de `create-cluster` a seguir cria um cluster.

```
aws ecs create-cluster \
  --cluster-name MyCluster
```

Saída:

```
{
  "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": []
  }
}
```



```
}
}
```

Para obter mais informações, consulte [Como criar um cluster](#) no Amazon ECS Developer Guide.

Exemplo 2: como criar um cluster usando provedores de capacidade

O exemplo de `create-cluster` a seguir cria um cluster e associa dois provedores de capacidade existentes a ele. O comando `create-capacity-provider` é usado para criar um provedor de capacidade. Embora recomendado, não é necessário especificar uma estratégia de provedor de capacidade. Neste exemplo, criamos um cluster chamado `MyCluster` e associamos os provedores de capacidade `MyCapacityProvider1` e `MyCapacityProvider2` a ele. Uma estratégia de provedor de capacidade padrão é especificada e distribui a tarefa igualmente em ambos os provedores de capacidade.

```
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
```

Saída:

```
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "PROVISIONING",
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [
      "MyCapacityProvider1",
      "MyCapacityProvider2"
    ],
    "defaultCapacityProviderStrategy": [
```

```

        {
            "capacityProvider": "MyCapacityProvider1",
            "weight": 1,
            "base": 0
        },
        {
            "capacityProvider": "MyCapacityProvider2",
            "weight": 1,
            "base": 0
        }
    ],
    "attachments": [
        {
            "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
            "type": "asp",
            "status": "PRECREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "MyCapacityProvider1"
                },
                {
                    "name": "scalingPlanName",
                    "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE111111"
                }
            ]
        },
        {
            "id": "ae592060-2382-4663-9476-b015c685593c",
            "type": "asp",
            "status": "PRECREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "MyCapacityProvider2"
                },
                {
                    "name": "scalingPlanName",
                    "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE222222"
                }
            ]
        }
    ]
}

```

```
    ],  
    "attachmentsStatus": "UPDATE_IN_PROGRESS"  
  }  
}
```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

Exemplo 3: como criar um cluster com várias tags

O exemplo `create-cluster` a seguir cria um cluster com várias tags. Para obter mais informações sobre como adicionar tags usando a sintaxe abreviada, consulte [Usando a sintaxe abreviada com a interface de linha de AWS comando no Guia do usuário](#).AWS CLI

```
aws ecs create-cluster \  
  --cluster-name MyCluster \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```

    ]
  }
}

```

Para obter mais informações, consulte [Como criar um cluster](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [CreateCluster](#) na Referência de AWS CLI Comandos.

create-service

O código de exemplo a seguir mostra como usar `create-service`.

AWS CLI

Exemplo 1: como criar um serviço com uma tarefa do Fargate

O exemplo de `create-service` a seguir demonstra como criar um serviço usando uma tarefa do 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

```

Saída:

```

{
  "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"
}
}
```

Exemplo 2: Para criar um serviço usando o tipo de EC2 inicialização

O `create-service` exemplo a seguir mostra como criar um serviço chamado `ecs-simple-service` com uma tarefa que usa o tipo de EC2 inicialização. O serviço usa a definição de tarefa `sleep360` e mantém uma instanciação da tarefa.

```
aws ecs create-service \
  --cluster MyCluster \
  --service-name ecs-simple-service \
  --task-definition sleep360:2 \
  --desired-count 1
```

Saída:

```
{
  "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",
    "enableECSTags": false,
    "propagateTags": "NONE"
  }
}
```

Exemplo 3: como criar um serviço que usa um controlador de implantação externo

O exemplo de `create-service` a seguir cria um serviço que usa um controlador de implantação externo.

```
aws ecs create-service \  
  --cluster MyCluster \  
  --service-name MyService \  
  --deployment-controller type=EXTERNAL \  
  --desired-count 1
```

Saída:

```
{  
  "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"
  }
}

```

Exemplo 4: como criar um serviço por trás de um balanceador de carga

O exemplo de `create-service` a seguir mostra como criar um serviço que fique por trás de um balanceador de carga. É necessário ter um balanceador de carga configurado na mesma região da instância de contêiner. Este exemplo usa a `--cli-input-json` opção e um arquivo JSON de entrada chamado `ecs-simple-service-elb.json` com o seguinte conteúdo:

```

{
  "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

```

Saída:

```

{
  "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
}
}

```

Para obter mais informações, consulte [Criação de um serviço](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [CreateService](#) na Referência de AWS CLI Comandos.

create-task-set

O código de exemplo a seguir mostra como usar create-task-set.

AWS CLI

Para criar um conjunto de tarefas

O create-task-set exemplo a seguir cria um conjunto de tarefas em um serviço que usa um controlador de implantação externo.

```
aws ecs create-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-definition MyTaskDefinition:2 \  
  --network-  
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

Saída:

```
{  
  "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  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [CreateTaskSet](#) na Referência de AWS CLI Comandos.

delete-account-setting

O código de exemplo a seguir mostra como usar `delete-account-setting`.

AWS CLI

Para excluir as configurações da conta de um IAM usuário ou IAM função específica

O exemplo a seguir `delete-account-setting` exclui as configurações da conta para o IAM usuário ou IAM função específica.

```
aws ecs delete-account-setting \  
  --name serviceLongArnFormat \  
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Saída:

```
{  
  "setting": {  
    "name": "serviceLongArnFormat",  
    "value": "enabled",  
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"  
  }  
}
```

Para obter mais informações, consulte [Amazon Resource Names \(ARNs\) e IDs](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DeleteAccountSetting](#) na Referência de AWS CLI Comandos.

delete-attributes

O código de exemplo a seguir mostra como usar `delete-attributes`.

AWS CLI

Para excluir um ou mais atributos personalizados de um ECS recurso da Amazon

O seguinte `delete-attributes` exclui um atributo com o nome `stack` de uma instância de contêiner.

```
aws ecs delete-attributes \  
  --attributes name=stack,targetId=arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Saída:

```
{  
  "attributes": [  
    {  
      "name": "stack",  
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",  
      "value": "production"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DeleteAttributes](#) na Referência de AWS CLI Comandos.

delete-capacity-provider

O código de exemplo a seguir mostra como usar `delete-capacity-provider`.

AWS CLI

Exemplo 1: Para excluir um provedor de capacidade usando o Amazon Resource Name (ARN)

O `delete-capacity-provider` exemplo a seguir exclui um provedor de capacidade especificando o Amazon Resource Name (ARN) do provedor de capacidade. ARN Assim como o status da exclusão do provedor de capacidade, podem ser recuperados usando o `describe-capacity-providers` comando.

```
aws ecs delete-capacity-provider \  
  --capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/  
ExampleCapacityProvider
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

Exemplo 2: Para excluir um provedor de capacidade usando o nome

O delete-capacity-provider exemplo a seguir exclui um provedor de capacidade especificando o nome abreviado do provedor de capacidade. O nome curto e o status da exclusão do provedor de capacidade podem ser recuperados usando o describe-capacity-providers comando.

```
aws ecs delete-capacity-provider \
  --capacity-provider ExampleCapacityProvider
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DeleteCapacityProvider](#) na Referência de AWS CLI Comandos.

delete-cluster

O código de exemplo a seguir mostra como usar delete-cluster.

AWS CLI

Como excluir um cluster vazio

O exemplo de delete-cluster a seguir exclui o cluster vazio especificado.

```
aws ecs delete-cluster --cluster MyCluster
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Excluindo um cluster](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DeleteCluster](#) na Referência de AWS CLI Comandos.

delete-service

O código de exemplo a seguir mostra como usar delete-service.

AWS CLI

Como excluir um serviço

O exemplo de `aws ecs delete-service` a seguir exclui o serviço especificado de um cluster. Inclua o parâmetro `--force` para excluir um serviço mesmo que ele não tenha sido reduzido a zero tarefas.

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

Para obter mais informações, consulte [Excluindo um serviço](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DeleteService](#) na Referência de AWS CLI Comandos.

delete-task-definitions

O código de exemplo a seguir mostra como usar delete-task-definitions.

AWS CLI

Para excluir uma definição de tarefa

O `delete-task-definitions` exemplo a seguir exclui uma definição de INACTIVE tarefa.


```
aws ecs delete-task-definitions \  
--task-definition curltest:1
```

Saída:

```
{  
  "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": []
}
```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DeleteTaskDefinitions](#) na Referência de AWS CLI Comandos.

delete-task-set

O código de exemplo a seguir mostra como usar `delete-task-set`.

AWS CLI

Para excluir um conjunto de tarefas

O `delete-task-set` exemplo a seguir mostra como excluir um conjunto de tarefas. Você pode incluir o `--force` parâmetro para excluir um conjunto de tarefas mesmo que ele não tenha sido escalado para 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
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [DeleteTaskSet](#) na Referência de AWS CLI Comandos.

deregister-container-instance

O código de exemplo a seguir mostra como usar `deregister-container-instance`.

AWS CLI

Para cancelar o registro de uma instância de contêiner de um cluster

O `deregister-container-instance` exemplo a seguir cancela o registro de uma instância de contêiner do cluster especificado. Se ainda houver tarefas em execução na instância do contêiner, você deverá interromper essas tarefas antes de cancelar o registro ou usar a opção.

`--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
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Cancelar o registro de uma instância de contêiner](#) no Guia do ECSdesenvolvedor.

- Para API obter detalhes, consulte [DeregisterContainerInstancena](#) Referência de AWS CLI Comandos.

deregister-task-definition

O código de exemplo a seguir mostra como usar `deregister-task-definition`.

AWS CLI

Para cancelar o registro de uma definição de tarefa

O `deregister-task-definition` exemplo a seguir cancela o registro da primeira revisão da definição da `curler` tarefa na sua região padrão.

```
aws ecs deregister-task-definition --task-definition curler:1
```

Observe que, na saída resultante, o status da definição da tarefa 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
}
```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DeregisterTaskDefinition](#) na Referência de AWS CLI Comandos.

describe-capacity-providers

O código de exemplo a seguir mostra como usar `describe-capacity-providers`.

AWS CLI

Exemplo 1: Para descrever todos os fornecedores de capacidade

O `describe-capacity-providers` exemplo a seguir recupera detalhes sobre todos os provedores de capacidade.

```
aws ecs describe-capacity-providers
```

Saída:

```
{
  "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": []
    }
]
}

```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

Exemplo 2: Para descrever um fornecedor de capacidade específico

O `describe-capacity-providers` exemplo a seguir recupera detalhes sobre um provedor de capacidade específico. O uso do `--include TAGS` parâmetro adicionará as tags associadas ao provedor de capacidade à saída.

```

aws ecs describe-capacity-providers \
  --capacity-providers MyCapacityProvider \

```

```
--include TAGS
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DescribeCapacityProviders](#) na Referência de AWS CLI Comandos.

describe-clusters

O código de exemplo a seguir mostra como usar `describe-clusters`.

AWS CLI

Exemplo 1: como descrever um cluster

O exemplo de `describe-clusters` a seguir recupera detalhes do cluster especificado.

```
aws ecs describe-clusters \  
  --cluster default
```

Saída:

```
{  
  "clusters": [  
    {  
      "status": "ACTIVE",  
      "clusterName": "default",  
      "registeredContainerInstancesCount": 0,  
      "pendingTasksCount": 0,  
      "runningTasksCount": 0,  
      "activeServicesCount": 1,  
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"  
    }  
  ],  
  "failures": []  
}
```

Para obter mais informações, consulte [Amazon ECS Clusters](#) no Amazon ECS Developer Guide.

Exemplo 2: como descrever um cluster com a opção de anexo

O `describe-clusters` exemplo a seguir especifica a `ATTACHMENTS` opção. Ele recupera detalhes sobre o cluster especificado e uma lista dos recursos anexados ao cluster no formato de anexos. Ao usar um provedor de capacidade com um cluster, os recursos, sejam `AutoScaling` planos ou políticas de escalabilidade, serão representados como `asp` ou `as_policy`.
`ATTACHMENTS`

```
aws ecs describe-clusters \  
  --include ATTACHMENTS \  
  --clusters sampleCluster
```

Saída:

```

{
  "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": []
}

```

Para obter mais informações, consulte [Amazon ECS Clusters](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DescribeClusters](#) na Referência de AWS CLI Comandos.

describe-container-instances

O código de exemplo a seguir mostra como usar `describe-container-instances`.

AWS CLI

Para descrever a instância do contêiner

O `describe-container-instances` exemplo a seguir recupera detalhes de uma instância de contêiner no update cluster, usando a instância de contêiner UUID como identificador.

```
aws ecs describe-container-instances \
  --cluster update \
  --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Saída:

```
{
  "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"
        }
    }
]
```

Para obter mais informações, consulte [Amazon ECS Container Instances](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DescribeContainerInstances](#) na Referência de AWS CLI Comandos.

describe-services

O código de exemplo a seguir mostra como usar `describe-services`.

AWS CLI

Para descrever um serviço

O `describe-services` exemplo a seguir recupera detalhes do `my-http-service` serviço no cluster padrão.

```
aws ecs describe-services --services my-http-service
```

Saída:

```
{
  "services": [
    {
      "status": "ACTIVE",
      "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/amazon-ecs-sample:1",
      "pendingCount": 0,
      "loadBalancers": [],
      "desiredCount": 10,
      "createdAt": 1466801808.595,
      "serviceName": "my-http-service",
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
```

```
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/my-http-
service",
    "deployments": [
      {
        "status": "PRIMARY",
        "pendingCount": 0,
        "createdAt": 1466801808.595,
        "desiredCount": 10,
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/amazon-ecs-sample:1",
        "updatedAt": 1428326312.703,
        "id": "ecs-svc/1234567890123456789",
        "runningCount": 10
      }
    ],
    "events": [
      {
        "message": "(service my-http-service) has reached a steady
state.",
        "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "createdAt": 1466801812.435
      }
    ],
    "runningCount": 10
  }
],
"failures": []
}
```

Para obter mais informações, consulte [Serviços](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DescribeServices](#) na Referência de AWS CLI Comandos.

describe-task-definition

O código de exemplo a seguir mostra como usar `describe-task-definition`.

AWS CLI

Para descrever uma definição de tarefa

O `describe-task-definition` exemplo a seguir recupera os detalhes de uma definição de tarefa.

```
aws ecs describe-task-definition \  
--task-definition hello_world:8
```

Saída:

```
{  
  "tasks": [  
    {  
      "attachments": [  
        {  
          "id": "17f3dff6-a9e9-4d83-99a9-7eb5193c2634",  
          "type": "ElasticNetworkInterface",  
          "status": "ATTACHED",  
          "details": [  
            {  
              "name": "subnetId",  
              "value": "subnet-0d0eab1bb38d5ca64"  
            },  
            {  
              "name": "networkInterfaceId",  
              "value": "eni-0d542ffb4a12aa6d9"  
            },  
            {  
              "name": "macAddress",  
              "value": "0e:6d:18:f6:2d:29"  
            },  
            {  
              "name": "privateDnsName",  
              "value": "ip-10-0-1-170.ec2.internal"  
            },  
            {  
              "name": "privateIPv4Address",  
              "value": "10.0.1.170"  
            }  
          ]  
        }  
      ],  
      "attributes": [  
        {  
          "name": "ecs.cpu-architecture",  
          "value": "x86_64"  
        }  
      ],  
    }  
  ]  
}
```

```
"availabilityZone": "us-east-1b",
"clusterArn": "arn:aws:ecs:us-east-1:053534965804:cluster/fargate-
cluster",
"connectivity": "CONNECTED",
"connectivityAt": "2023-11-28T11:10:52.907000-05:00",
"containers": [
  {
    "containerArn": "arn:aws:ecs:us-east-1:053534965804:container/
fargate-cluster/
c524291ae4154100b601a543108b193a/772c4784-92ae-414e-8df2-03d3358e39fa",
    "taskArn": "arn:aws:ecs:us-east-1:053534965804:task/fargate-
cluster/c524291ae4154100b601a543108b193a",
    "name": "web",
    "image": "nginx",
    "imageDigest":
"sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee",
    "runtimeId": "c524291ae4154100b601a543108b193a-265927825",
    "lastStatus": "RUNNING",
    "networkBindings": [],
    "networkInterfaces": [
      {
        "attachmentId": "17f3dff6-a9e9-4d83-99a9-7eb5193c2634",
        "privateIpv4Address": "10.0.1.170"
      }
    ],
    "healthStatus": "HEALTHY",
    "cpu": "99",
    "memory": "100"
  },
  {
    "containerArn": "arn:aws:ecs:us-east-1:053534965804:container/
fargate-cluster/c524291ae4154100b601a543108b193a/c051a779-40d2-48ca-
ad5e-6ec875ceb610",
    "taskArn": "arn:aws:ecs:us-east-1:053534965804:task/fargate-
cluster/c524291ae4154100b601a543108b193a",
    "name": "aws-guardduty-agent-FvWGoDU",
    "imageDigest":
"sha256:359b8b014e5076c625daa1056090e522631587a7afa3b2e055edda6bd1141017",
    "runtimeId": "c524291ae4154100b601a543108b193a-505093495",
    "lastStatus": "RUNNING",
    "networkBindings": [],
    "networkInterfaces": [
      {
        "attachmentId": "17f3dff6-a9e9-4d83-99a9-7eb5193c2634",
```

```

        "privateIpv4Address": "10.0.1.170"
      }
    ],
    "healthStatus": "UNKNOWN"
  }
],
"cpu": "256",
"createdAt": "2023-11-28T11:10:49.299000-05:00",
"desiredStatus": "RUNNING",
"enableExecuteCommand": false,
"group": "family:webserver",
"healthStatus": "HEALTHY",
"lastStatus": "RUNNING",
"launchType": "FARGATE",
"memory": "512"
"platformVersion": "1.4.0",
"platformFamily": "Linux",
"pullStartedAt": "2023-11-28T11:10:59.773000-05:00",
"pullStoppedAt": "2023-11-28T11:11:12.624000-05:00",
"startedAt": "2023-11-28T11:11:20.316000-05:00",
"tags": [],
"taskArn": "arn:aws:ecs:us-east-1:053534965804:task/fargate-cluster/
c524291ae4154100b601a543108b193a",
"taskDefinitionArn": "arn:aws:ecs:us-east-1:053534965804:task-
definition/webserver:5",
"version": 4,
"ephemeralStorage": {
  "sizeInGiB": 20
}
}
],
"failures": []
}

```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DescribeTaskDefinition](#) na Referência de AWS CLI Comandos.

describe-task-sets

O código de exemplo a seguir mostra como usar `describe-task-sets`.

AWS CLI

Para descrever um conjunto de tarefas

O `describe-task-sets` exemplo a seguir descreve um conjunto de tarefas em um serviço que usa um implantador externo.

```
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
```

Saída:

```
{  
  "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": []
}

```

- Para API obter detalhes, consulte [DescribeTaskSets](#) na Referência de AWS CLI Comandos.

describe-tasks

O código de exemplo a seguir mostra como usar `describe-tasks`.

AWS CLI

Exemplo 1: como descrever uma única tarefa

O exemplo de `describe-tasks` a seguir recupera os detalhes de uma tarefa em um cluster. Você pode especificar a tarefa usando a ID ou a totalidade ARN da tarefa. Este exemplo usa a totalidade ARN da tarefa.

```

aws ecs describe-tasks \
  --cluster MyCluster \
  --tasks arn:aws:ecs:us-east-1:123456789012:task/MyCluster/4d590253bb114126b7afa7b58EXAMPLE

```

Saída:

```

{
  "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": []
}

```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

Exemplo 2: como descrever várias tarefas

O exemplo de `describe-tasks` a seguir recupera os detalhes de várias tarefas de um cluster. Você pode especificar a tarefa usando a ID ou a totalidade ARN da tarefa. Este exemplo usa todas IDs as tarefas.

```

aws ecs describe-tasks \
  --cluster MyCluster \
  --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"

```

Saída:

```

{
  "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": []  
}
```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [DescribeTasks](#) na Referência de AWS CLI Comandos.

execute-command

O código de exemplo a seguir mostra como usar `execute-command`.

AWS CLI

Para executar um comando `/bin/sh` interativo

O `execute-command` exemplo a seguir executa um comando `/bin/sh` interativo em um contêiner nomeado `MyContainer` para uma tarefa com um id de `arn:aws:ecs:us-east-1:123456789012:task/MyCluster/d789e94343414c25b9f6bd59eEXAMPLE`

```
aws ecs execute-command \  
  --cluster MyCluster \  
  --task arn:aws:ecs:us-east-1:123456789012:task/MyCluster/  
d789e94343414c25b9f6bd59eEXAMPLE \  
  --container MyContainer \  
  --interactive \  
  --command "/bin/sh"
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o Amazon ECS Exec para depuração no Amazon Developer Guide](#). ECS

- Para API obter detalhes, consulte [ExecuteCommand](#) na Referência de AWS CLI Comandos.

list-account-settings

O código de exemplo a seguir mostra como usar `list-account-settings`.

AWS CLI

Exemplo 1: Para ver as configurações de conta de uma conta

O `list-account-settings` exemplo a seguir exibe as configurações efetivas da conta.

```
aws ecs list-account-settings --effective-settings
```

Saída:

```
{
  "settings": [
    {
      "name": "containerInstanceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    },
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    },
    {
      "name": "taskLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:root"
    }
  ]
}
```

Exemplo 2: Para visualizar as configurações da conta de um IAM usuário ou IAM função específica

O `list-account-settings` exemplo a seguir exibe as configurações da conta para o IAM usuário ou IAM função especificada.

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```


Saída:

```
{
  "settings": [
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
  ]
}
```

Para obter mais informações, consulte [Amazon Resource Names \(ARNs\) e IDs](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListAccountSettings](#) na Referência de AWS CLI Comandos.

list-attributes

O código de exemplo a seguir mostra como usar `list-attributes`.

AWS CLI

Para listar as instâncias do contêiner que contêm um atributo específico

O exemplo a seguir lista os atributos das instâncias de contêiner que têm o `stack=production` atributo no cluster padrão.

```
aws ecs list-attributes \
  --target-type container-instance \
  --attribute-name stack \
  --attribute-value production \
  --cluster default
```

Saída:

```
{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
```

```
        "value": "production"
      }
    ]
  }
```

Para obter mais informações, consulte [Configuração do Amazon ECS Container Agent](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListAttributes](#) na Referência de AWS CLI Comandos.

list-clusters

O código de exemplo a seguir mostra como usar `list-clusters`.

AWS CLI

Como listar os clusters disponíveis

O exemplo de `list-clusters` a seguir lista todos os clusters disponíveis.

```
aws ecs list-clusters
```

Saída:

```
{
  "clusterArns": [
    "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
    "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
  ]
}
```

Para obter mais informações, consulte [Amazon ECS Clusters](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListClusters](#) na Referência de AWS CLI Comandos.

list-container-instances

O código de exemplo a seguir mostra como usar `list-container-instances`.

AWS CLI

Para listar as instâncias do contêiner em um cluster

O `list-container-instances` exemplo a seguir lista todas as instâncias de contêiner disponíveis em um cluster.

```
aws ecs list-container-instances --cluster MyCluster
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Amazon ECS Container Instances](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListContainerInstances](#) na Referência de AWS CLI Comandos.

list-services-by-namespace

O código de exemplo a seguir mostra como usar `list-services-by-namespace`.

AWS CLI

Para listar os serviços em um namespace

O `list-services-by-namespace` exemplo a seguir lista todos os serviços configurados para o namespace especificado em sua região padrão.

```
aws ecs list-services-by-namespace \
  --namespace service-connect
```

Saída:

```
{
  "serviceArns": [
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",
  ]
}
```

```
    "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-  
service"  
  ]  
}
```

Para obter mais informações, consulte [Service Connect](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListServicesByNamespace](#) na Referência de AWS CLI Comandos.

list-services

O código de exemplo a seguir mostra como usar `list-services`.

AWS CLI

Como listar os serviços em um cluster

O exemplo de `list-services` a seguir mostra como listar os serviços em execução em um cluster.

```
aws ecs list-services --cluster MyCluster
```

Saída:

```
{  
  "serviceArns": [  
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"  
  ]  
}
```

Para obter mais informações, consulte [Serviços](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListServices](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags de um cluster específico.

```
aws ecs list-tags-for-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

Saída:

```
{  
  "tags": [  
    {  
      "key": "key1",  
      "value": "value1"  
    },  
    {  
      "key": "key2",  
      "value": "value2"  
    },  
    {  
      "key": "key3",  
      "value": "value3"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-task-definition-families

O código de exemplo a seguir mostra como usar `list-task-definition-families`.

AWS CLI

Exemplo 1: Para listar as famílias de definição de tarefas registradas

O `list-task-definition-families` exemplo a seguir lista todas as famílias de definição de tarefas registradas.

```
aws ecs list-task-definition-families
```

Saída:

```
{
  "families": [
    "node-js-app",
    "web-timer",
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

Exemplo 2: Para filtrar as famílias de definição de tarefas registradas

O `list-task-definition-families` exemplo a seguir lista as revisões de definição de tarefas que começam com “hpcc”.

```
aws ecs list-task-definition-families --family-prefix hpcc
```

Saída:

```
{
  "families": [
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

Para obter mais informações, consulte [Parâmetros de definição de tarefas](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListTaskDefinitionFamilies](#) na Referência de AWS CLI Comandos.

list-task-definitions

O código de exemplo a seguir mostra como usar `list-task-definitions`.

AWS CLI

Exemplo 1: Para listar as definições de tarefas registradas

O `list-task-definitions` exemplo a seguir lista todas as definições de tarefas registradas.

```
aws ecs list-task-definitions
```

Saída:

```
{
  "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"
  ]
}
```

Exemplo 2: Para listar as definições de tarefas registradas em uma família

O `list-task-definitions` exemplo a seguir lista as revisões da definição de tarefas de uma família especificada.

```
aws ecs list-task-definitions --family-prefix wordpress
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListTaskDefinitions](#) na Referência de AWS CLI Comandos.

list-tasks

O código de exemplo a seguir mostra como usar `list-tasks`.

AWS CLI

Exemplo 1: como listar as tarefas em um cluster

O exemplo de `list-tasks` a seguir lista todas as tarefas de um cluster.

```
aws ecs list-tasks --cluster default
```

Saída:

```
{
  "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"
  ]
}
```

Exemplo 2: como listar as tarefas de uma instância de contêiner específica

O `list-tasks` exemplo a seguir lista as tarefas em uma instância de contêiner, usando a instância de contêiner UUID como filtro.

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

Saída:

```
{
  "taskArns": [
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE"
  ]
}
```

Para obter mais informações, consulte [Amazon ECS Task Definitions](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [ListTasks](#) na Referência de AWS CLI Comandos.

put-account-setting-default

O código de exemplo a seguir mostra como usar `put-account-setting-default`.

AWS CLI

Para modificar as configurações padrão da conta

O `put-account-setting-default` exemplo a seguir modifica a configuração padrão da conta para todos os IAM usuários ou funções em sua conta. Essas alterações se aplicam a toda a AWS conta, a menos que um IAM usuário ou função substitua explicitamente essas configurações por si mesmos.

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

Saída:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:root"
  }
}
```

Para obter mais informações, consulte [Amazon Resource Names \(ARNs\) e IDs](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [PutAccountSettingDefault](#) na Referência de AWS CLI Comandos.

put-account-setting

O código de exemplo a seguir mostra como usar `put-account-setting`.

AWS CLI

Para modificar a configuração da conta da sua conta de IAM usuário

O `put-account-setting` exemplo a seguir ativa a configuração da `serviceLongArnFormat` conta para sua conta de IAM usuário.

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled
```

Saída:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::130757420319:user/your_username"
  }
}
```

Para obter mais informações, consulte [Modificação das configurações da conta](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [PutAccountSetting](#) na Referência de AWS CLI Comandos.

put-account-settings

O código de exemplo a seguir mostra como usar `put-account-settings`.

AWS CLI

Para modificar as configurações da conta de um IAM usuário ou IAM função

O `put-account-setting` exemplo a seguir modifica as configurações da conta para o IAM usuário ou IAM função especificada.

```
aws ecs put-account-setting \
  --name serviceLongArnFormat \
  --value enabled \
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Saída:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"
  }
}
```

```
}
```

- Para API obter detalhes, consulte [PutAccountSettings](#) na Referência de AWS CLI Comandos.

put-attributes

O código de exemplo a seguir mostra como usar `put-attributes`.

AWS CLI

Para criar um atributo e associá-lo a um ECS recurso da Amazon

O seguinte `put-attributes` aplica um atributo com o nome `stack` e a produção de valor a uma instância de contêiner.

```
aws ecs put-attributes \  
  --attributes name=stack,value=production,targetId=arn:aws:ecs:us-  
west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Saída:

```
{  
  "attributes": [  
    {  
      "name": "stack",  
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",  
      "value": "production"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [PutAttributes](#) na Referência de AWS CLI Comandos.

put-cluster-capacity-providers

O código de exemplo a seguir mostra como usar `put-cluster-capacity-providers`.

AWS CLI

Exemplo 1: Para adicionar um provedor de capacidade existente a um cluster

O `put-cluster-capacity-providers` exemplo a seguir adiciona um provedor de capacidade existente a um cluster. O comando `create-capacity-provider` é usado para criar um provedor de capacidade. O `describe-clusters` comando é usado para descrever os provedores de capacidade atuais e a estratégia padrão do provedor de capacidade associada a um cluster. Ao adicionar um novo provedor de capacidade a um cluster, você deve especificar todos os provedores de capacidade existentes, além do novo provedor de capacidade que você deseja associar ao cluster. Você também deve especificar a estratégia padrão do provedor de capacidade a ser associada ao cluster. Neste exemplo, o `MyCluster` cluster tem o provedor de `MyCapacityProvider1` capacidade associado a ele e você deseja adicionar o provedor de `MyCapacityProvider2` capacidade e incluí-lo na estratégia padrão do provedor de capacidade para que as tarefas sejam distribuídas uniformemente entre os dois provedores de capacidade.

```
aws ecs put-cluster-capacity-providers \  
  --cluster MyCluster \  
  --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \  
  --default-capacity-provider-  
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

Saída:

```
{  
  "cluster": {  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",  
    "clusterName": "MyCluster",  
    "status": "ACTIVE",  
    "registeredContainerInstancesCount": 0,  
    "runningTasksCount": 0,  
    "pendingTasksCount": 0,  
    "activeServicesCount": 0,  
    "statistics": [],  
    "tags": [],  
    "settings": [  
      {  
        "name": "containerInsights",  
        "value": "enabled"  
      }  
    ],  
    "capacityProviders": [  
      "MyCapacityProvider1",  
      "MyCapacityProvider2"  
    ],  
  },  
}
```

```
"defaultCapacityProviderStrategy": [
  {
    "capacityProvider": "MyCapacityProvider1",
    "weight": 1,
    "base": 0
  },
  {
    "capacityProvider": "MyCapacityProvider2",
    "weight": 1,
    "base": 0
  }
],
"attachments": [
  {
    "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
    "type": "as_policy",
    "status": "ACTIVE",
    "details": [
      {
        "name": "capacityProviderName",
        "value": "MyCapacityProvider1"
      },
      {
        "name": "scalingPolicyName",
        "value": "ECManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
      }
    ]
  },
  {
    "id": "ae592060-2382-4663-9476-b015c685593c",
    "type": "as_policy",
    "status": "ACTIVE",
    "details": [
      {
        "name": "capacityProviderName",
        "value": "MyCapacityProvider2"
      },
      {
        "name": "scalingPolicyName",
        "value": "ECManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
      }
    ]
  }
]
```

```
    }
  ],
  "attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}
```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

Exemplo 2: Para remover um provedor de capacidade de um cluster

O `put-cluster-capacity-providers` exemplo a seguir remove um provedor de capacidade de um cluster. O `describe-clusters` comando é usado para descrever os provedores de capacidade atuais associados a um cluster. Ao remover um provedor de capacidade de um cluster, você deve especificar os provedores de capacidade que deseja que permaneçam associados ao cluster, bem como a estratégia padrão do provedor de capacidade a ser associada ao cluster. Neste exemplo, o cluster tem os provedores de `MyCapacityProvider2` capacidade `MyCapacityProvider1` e associados a ele e você deseja remover o provedor de `MyCapacityProvider2` capacidade, então você especifica somente `MyCapacityProvider1` no comando junto com a estratégia atualizada do provedor de capacidade padrão.

```
aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers MyCapacityProvider1 \
  --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0
```

Saída:

```
{
  "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": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
        }
      ]
    },
    {
      "id": "ae592060-2382-4663-9476-b015c685593c",
      "type": "as_policy",
      "status": "DELETING",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider2"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
        }
      ]
    }
  ]
}
```

```

    ]
  }
],
"attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

Exemplo 3: Para remover todos os provedores de capacidade de um cluster

O `put-cluster-capacity-providers` exemplo a seguir remove todos os provedores de capacidade existentes do cluster.

```

aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers [] \
  --default-capacity-provider-strategy []

```

Saída:

```

{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
    "clusterName": "MyCluster",
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "enabled"
      }
    ],
    "capacityProviders": [],
    "defaultCapacityProviderStrategy": [],
    "attachments": [

```



```

    {
      "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
      "type": "as_policy",
      "status": "DELETING",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider1"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE111111"
        }
      ]
    },
    {
      "id": "ae592060-2382-4663-9476-b015c685593c",
      "type": "as_policy",
      "status": "DELETING",
      "details": [
        {
          "name": "capacityProviderName",
          "value": "MyCapacityProvider2"
        },
        {
          "name": "scalingPolicyName",
          "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE222222"
        }
      ]
    }
  ],
  "attachmentsStatus": "UPDATE_IN_PROGRESS"
}

```

Para obter mais informações, consulte [Provedores de capacidade de cluster](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [PutClusterCapacityProviders](#) na Referência de AWS CLI Comandos.

register-task-definition

O código de exemplo a seguir mostra como usar `register-task-definition`.

AWS CLI

Exemplo 1: Para registrar uma definição de tarefa com um JSON arquivo

O `register-task-definition` exemplo a seguir registra uma definição de tarefa na família especificada. As definições do contêiner são salvas em JSON formato no local do arquivo especificado.

```
aws ecs register-task-definition \  
  --cli-input-json file://<path_to_json_file>/sleep360.json
```

Conteúdo de `sleep360.json`:

```
{  
  "containerDefinitions": [  
    {  
      "name": "sleep",  
      "image": "busybox",  
      "cpu": 10,  
      "command": [  
        "sleep",  
        "360"  
      ],  
      "memory": 10,  
      "essential": true  
    }  
  ],  
  "family": "sleep360"  
}
```

Saída:

```
{  
  "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
}

```

Para obter mais informações, consulte [Exemplos de definições de tarefas](#) no Amazon ECS Developer Guide.

Exemplo 2: Para registrar uma definição de tarefa com um parâmetro de JSON string

O `register-task-definition` exemplo a seguir registra uma definição de tarefa usando definições de contêiner fornecidas como um parâmetro de JSON string com aspas duplas de escape.

```

aws ecs register-task-definition \
  --family sleep360 \
  --container-definitions "[{\"name\":\"sleep\",\"image\":\"busybox\",\"cpu\":10,
  \command\":[\"sleep\",\"360\"],\"memory\":10,\"essential\":true}]"

```

A saída é idêntica à do exemplo anterior.

Para obter mais informações, consulte [Criação de uma definição de tarefa](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [RegisterTaskDefinition](#) na Referência de AWS CLI Comandos.

run-task

O código de exemplo a seguir mostra como usar `run-task`.

AWS CLI

Para executar uma tarefa em seu cluster padrão

O `run-task` exemplo a seguir executa uma tarefa no cluster padrão e usa um token de cliente.

```
aws ecs run-task \  
  --cluster default \  
  --task-definition sleep360:1 \  
  --client-token 550e8400-e29b-41d4-a716-446655440000
```

Saída:

```
{  
  "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": []
}
```

Para obter mais informações, consulte [Executando tarefas](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [RunTask](#) na Referência de AWS CLI Comandos.

start-task

O código de exemplo a seguir mostra como usar `start-task`.

AWS CLI

Para iniciar uma nova tarefa

O seguinte `start-task` inicia uma tarefa usando a revisão mais recente da definição da `sleep360` tarefa na instância de contêiner especificada no cluster padrão.

```
aws ecs start-task \  
  --task-definition sleep360 \  
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

Saída:

```
{  
  "tasks": [  
    {  
      "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/  
default/666fdccc2e2d4b6894dd422f4eeee8f8",  
      "clusterArn": "arn:aws:ecs:us-west-2:130757420319:cluster/default",  
      "taskDefinitionArn": "arn:aws:ecs:us-west-2:130757420319:task-  
definition/sleep360:3",  
      "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/default/765936fadbdd46b5991a4bd70c2a43d4",  
      "overrides": {  
        "containerOverrides": [  
          {  
            "name": "sleep"  
          }  
        ]  
      },  
      "lastStatus": "PENDING",  
      "desiredStatus": "RUNNING",  
      "cpu": "128",  
      "memory": "128",  
      "containers": [  
        {  
          "containerArn": "arn:aws:ecs:us-  
west-2:130757420319:container/75f11ed4-8a3d-4f26-a33b-ad1db9e02d41",
```

```

        "taskArn": "arn:aws:ecs:us-west-2:130757420319:task/
default/666fdccc2e2d4b6894dd422f4eeee8f8",
        "name": "sleep",
        "lastStatus": "PENDING",
        "networkInterfaces": [],
        "cpu": "10",
        "memory": "10"
    }
],
"version": 1,
"createdAt": 1563421494.186,
"group": "family:sleep360",
"launchType": "EC2",
"attachments": [],
"tags": []
}
],
"failures": []
}

```

- Para API obter detalhes, consulte [StartTask](#) na Referência de AWS CLI Comandos.

stop-task

O código de exemplo a seguir mostra como usar stop-task.

AWS CLI

Para interromper uma tarefa

O seguinte stop-task impede que a tarefa especificada seja executada no cluster padrão.

```

aws ecs stop-task \
  --task 666fdccc2e2d4b6894dd422f4eeee8f8

```

Saída:

```

{
  "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": []
}
}

```

- Para API obter detalhes, consulte [StopTask](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar um recurso

O tag-resource exemplo a seguir adiciona uma única tag ao recurso especificado.

```
aws ecs tag-resource \
```



```
--resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
--tags key=key1,value=value1
```

Este comando não produz saída.

Para adicionar várias tags a um recurso

O tag-resource exemplo a seguir adiciona várias tags ao recurso especificado.

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover uma tag de um recurso

O untag-resource exemplo a seguir remove as tags listadas do recurso especificado.

```
aws ecs untag-resource \
--resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
--tag-keys key1,key2
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-cluster-settings

O código de exemplo a seguir mostra como usar update-cluster-settings.

AWS CLI

Para modificar as configurações do seu cluster

O `update-cluster-settings` exemplo a seguir ativa o CloudWatch Container Insights para o `default` cluster.

```
aws ecs update-cluster-settings \  
  --cluster default \  
  --settings name=containerInsights,value=enabled
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Modificação das configurações da conta](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [UpdateClusterSettings](#) na Referência de AWS CLI Comandos.

update-container-agent

O código de exemplo a seguir mostra como usar `update-container-agent`.

AWS CLI

Para atualizar o agente de contêiner em uma instância de ECS contêiner da Amazon

O `update-container-agent` exemplo a seguir atualiza o agente de contêiner na instância de contêiner especificada no cluster padrão.

```
aws ecs update-container-agent --cluster default --container-  
instance a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

Saída:

```
{  
  "containerInstance": {  
    "status": "ACTIVE",  
    ...  
    "agentUpdateStatus": "PENDING",  
    "versionInfo": {  
      "agentVersion": "1.0.0",  
      "agentHash": "4023248",  
      "dockerVersion": "DockerVersion: 1.5.0"  
    }  
  }  
}
```

Para obter mais informações, consulte [Atualização do Amazon ECS Container Agent](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [UpdateContainerAgent](#) na Referência de AWS CLI Comandos.

update-container-instances-state

O código de exemplo a seguir mostra como usar `update-container-instances-state`.

AWS CLI

Para atualizar o estado de uma instância de contêiner

O seguinte `update-container-instances-state` atualiza o estado da instância de contêiner especificada para a `DRAINING` qual a removerá do cluster em que está registrada.

```
aws ecs update-container-instances-state \  
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \  
  --status DRAINING
```

Saída:

```
{
  "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": []
```

```
}
```

- Para API obter detalhes, consulte [UpdateContainerInstancesState](#) na Referência de AWS CLI Comandos.

update-service-primary-task-set

O código de exemplo a seguir mostra como usar `update-service-primary-task-set`.

AWS CLI

Para atualizar o conjunto de tarefas principal de um serviço

O `update-service-primary-task-set` exemplo a seguir atualiza o conjunto de tarefas principal para o serviço especificado.

```
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
```

Saída:

```
{  
  "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
}
}
```

- Para API obter detalhes, consulte [UpdateServicePrimaryTaskSet](#) na Referência de AWS CLI Comandos.

update-service

O código de exemplo a seguir mostra como usar `update-service`.

AWS CLI

Exemplo 1: como alterar a definição de tarefa usada em um serviço

O exemplo de `update-service` a seguir atualiza o serviço `my-http-service` para usar a definição de tarefa `amazon-ecs-sample`.

```
aws ecs update-service --service my-http-service --task-definition amazon-ecs-sample
```

Exemplo 2: Para alterar o número de tarefas em um serviço

O exemplo de `update-service` a seguir atualiza a contagem de tarefas desejada do serviço `my-http-service` para 3.

```
aws ecs update-service --service my-http-service --desired-count 3
```

Para obter mais informações, consulte [Atualização de um serviço](#) no Amazon ECS Developer Guide.

- Para API obter detalhes, consulte [UpdateService](#) na Referência de AWS CLI Comandos.

update-task-set

O código de exemplo a seguir mostra como usar `update-task-set`.

AWS CLI

Para atualizar um conjunto de tarefas

O `update-task-set` exemplo a seguir atualiza um conjunto de tarefas para ajustar a escala.

```
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
```

Saída:

```
{
  "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
}
}
```

- Para API obter detalhes, consulte [UpdateTaskSet](#) na Referência de AWS CLI Comandos.

EFSExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonEFS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-file-system

O código de exemplo a seguir mostra como usar `create-file-system`.

AWS CLI

Para criar um sistema de arquivos criptografado

O `create-file-system` exemplo a seguir cria um sistema de arquivos criptografado usando o padrão CMK. Ele também adiciona a `tagName=my-file-system`.

```
aws efs create-file-system \  
  --performance-mode generalPurpose \  
  --throughput-mode bursting \  
  --encrypted \  
  --tags Key=Name,Value=my-file-system
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Criação de sistemas de EFS arquivos da Amazon](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [CreateFileSystem](#) na Referência de AWS CLI Comandos.

create-mount-target

O código de exemplo a seguir mostra como usar create-mount-target.

AWS CLI

Para criar um alvo de montagem

O create-mount-target exemplo a seguir cria um destino de montagem para o sistema de arquivos especificado.

```
aws efs create-mount-target \  
  --file-system-id fs-c7a0456e \  
  --subnet-id subnet-02bf4c428bexample \  
  --security-groups sg-068f739363example
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Criação de destinos de montagem](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [CreateMountTarget](#) na Referência de AWS CLI Comandos.

delete-file-system

O código de exemplo a seguir mostra como usar `delete-file-system`.

AWS CLI

Para excluir um sistema de arquivos

O `delete-file-system` exemplo a seguir exclui o sistema de arquivos especificado.

```
aws efs delete-file-system \  
  --file-system-id fs-c7a0456e
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um sistema de EFS arquivos da Amazon](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [DeleteFileSystem](#) na Referência de AWS CLI Comandos.

delete-mount-target

O código de exemplo a seguir mostra como usar `delete-mount-target`.

AWS CLI

Para excluir um alvo de montagem

O `delete-mount-target` exemplo a seguir exclui o destino de montagem especificado.

```
aws efs delete-mount-target \  
  --mount-target-id fsmt-f9a14450
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de destinos de montagem](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [DeleteMountTarget](#) na Referência de AWS CLI Comandos.

describe-file-systems

O código de exemplo a seguir mostra como usar `describe-file-systems`.

AWS CLI

Para descrever um sistema de arquivos

O `describe-file-systems` exemplo a seguir descreve o sistema de arquivos especificado.

```
aws efs describe-file-systems \  
  --file-system-id fs-c7a0456e
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciamento de sistemas de EFS arquivos da Amazon](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [DescribeFileSystems](#) na Referência de AWS CLI Comandos.

describe-mount-targets

O código de exemplo a seguir mostra como usar `describe-mount-targets`.

AWS CLI

Para descrever um alvo de montagem

O `describe-mount-targets` exemplo a seguir descreve o destino de montagem especificado.

```
aws efs describe-mount-targets \  
  --mount-target-id fsmt-f9a14450
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Criação de destinos de montagem](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [DescribeMountTargets](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever as tags de um sistema de arquivos

O `describe-tags` exemplo a seguir descreve as tags do sistema de arquivos especificado.

```
aws efs describe-tags \  
  --file-system-id fs-c7a0456e
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    },  
    {  
      "Key": "Department",  
      "Value": "Business Intelligence"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciamento de tags do sistema de arquivos](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para recuperar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir recupera as tags associadas ao sistema de arquivos especificado.

```
aws efs list-tags-for-resource \  
  --resource-id fs-c7a0456e
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    },  
    {  
      "Key": "Department",  
      "Value": "Business Intelligence"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciamento de tags do sistema de arquivos](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir adiciona a tag `Department=Business Intelligence` ao sistema de arquivos especificado.

```
aws efs tag-resource \  
  --resource-id fs-c7a0456e \  
  --tags Key=Department,Value="Business Intelligence"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags do sistema de arquivos](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove a tag com a chave da `Department` tag do sistema de arquivos especificado.

```
aws efs untag-resource \  
  --resource-id fs-c7a0456e \  
  --tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags do sistema de arquivos](#) no Guia do usuário do Amazon Elastic File System.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

EKS Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonEKS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-encryption-config

O código de exemplo a seguir mostra como usar `associate-encryption-config`.

AWS CLI

Para associar uma configuração de criptografia a um cluster existente

O `associate-encryption-config` exemplo a seguir habilita a criptografia em EKS clusters existentes que ainda não têm a criptografia ativada.

```
aws eks associate-encryption-config \
  --cluster-name my-eks-cluster \
  --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Habilitar a criptografia secreta em um cluster existente](#) no Guia EKS do usuário da Amazon.

- Para API obter detalhes, consulte [AssociateEncryptionConfig](#) na Referência de AWS CLI Comandos.

associate-identity-provider-config

O código de exemplo a seguir mostra como usar `associate-identity-provider-config`.

AWS CLI

Associe o provedor de identidade ao seu Amazon EKS Cluster

O `associate-identity-provider-config` exemplo a seguir associa um provedor de identidade ao seu 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=m
username-prefix,groupsClaim=my-claim,groupsPrefix=my-groups-
prefix,requiredClaims={Claim1=value1,Claim2=value2}' \
  --tags env=dev
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Autenticar usuários do seu cluster a partir de um provedor de identidade do OpenID Connect - OIDC Associar um provedor](#) de identidade no Guia do usuário da EKS Amazon.

- Para API obter detalhes, consulte [AssociateIdentityProviderConfig](#) Referência de AWS CLI Comandos.

create-addon

O código de exemplo a seguir mostra como usar create-addon.

AWS CLI

Exemplo 1: Para criar um EKS complemento da Amazon com uma versão compatível padrão para a respectiva versão do cluster EKS

O comando de create-addon exemplo a seguir cria um EKS complemento da Amazon com versão compatível padrão para a respectiva versão do 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
```

Saída:

```
{  
  "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": {}  
  }  
}
```



```
}
```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criando um complemento no Guia EKS](#) do usuário da Amazon.

Exemplo 2: Para criar um EKS complemento da Amazon com uma versão específica do complemento

O comando de `create-addon` exemplo a seguir cria um EKS complemento da Amazon com uma versão específica do complemento.

```
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
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criando um complemento no Guia EKS](#) do usuário da Amazon.

Exemplo 3: Para criar um EKS complemento da Amazon com valores de configuração personalizados e resolver detalhes de conflitos

O comando de `create-addon` exemplo a seguir cria um EKS complemento da Amazon com valores de configuração personalizados e resolve detalhes de conflitos.

```
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
```

Saída:

```
{
  "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\"}}}"
  }
}
```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criação de um complemento no Guia EKS do usuário da Amazon](#).

Exemplo 4: Para criar um EKS complemento da Amazon com arquivo de valores JSON de configuração personalizados

O comando de `create-addon` exemplo a seguir cria um EKS complemento da Amazon com valores de configuração personalizados e detalhes de resolução de conflitos.

```
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"}'
```

Conteúdo de `configuration-values.json`:

```
{  
  "resources": {  
    "limits": {  
      "cpu": "150m"  
    }  
  },  
  "env": {  
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"  
  }  
}
```

Saída:

```
{  
  "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-fb32-6a7be19eecd",  
    "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
  {\n
    \"resources\": {\n
      \"limits\": {\n
        \"cpu\": \"150m\"\n
      }\n
    },\n
    \"env\": {\n
      \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n
    }\n
  }
}"
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criação de um complemento no](#) Guia EKS do usuário da Amazon.

Exemplo 5: Para criar um EKS complemento da Amazon com arquivo de valores YAML de configuração personalizados

O comando de create-addon exemplo a seguir cria um EKS complemento da Amazon com valores de configuração personalizados e detalhes de resolução de conflitos.

```

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"}'

```

Conteúdo de configuration-values.yaml:

```

resources:
  limits:
    cpu: '100m'
env:
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'

```

Saída:

```

{
  "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'"
    }
  }
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criação de um complemento no](#) Guia EKS do usuário da Amazon.

- Para API obter detalhes, consulte [CreateAddon](#) na Referência de AWS CLI Comandos.

create-cluster

O código de exemplo a seguir mostra como usar `create-cluster`.

AWS CLI

Como criar um cluster

Este comando de exemplo cria o cluster `prod` na sua região padrão.

Comando:

```

aws eks create-cluster --name prod \
--role-arn arn:aws:iam::012345678910:role/eks-service-role-
AWSServiceRoleForAmazonEKS-J7ONKE3BQ4PI \
--resources-vpc-config subnetIds=subnet-6782e71e,subnet-
e7e761ac,securityGroupIds=sg-6979fe18

```

Saída:

```
{
```

```

"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": {}
}
}

```

Como criar um cluster com acesso ao endpoint privado e registro em log habilitado

Este comando de exemplo cria o cluster `example` na região padrão com acesso ao endpoint público desabilitado e acesso ao endpoint privado e todos os tipos de registro em log habilitados.

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}]}'

```

Saída:

```

{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateCluster](#) na Referência de AWS CLI Comandos.

create-fargate-profile

O código de exemplo a seguir mostra como usar `create-fargate-profile`.

AWS CLI

Exemplo 1: Criar perfil EKS Fargate para um seletor com um namespace

O `create-fargate-profile` exemplo a seguir cria um Perfil EKS Fargate para um seletor com um 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"}]'
```

Saída:

```
{  
  "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": {}  
  }  
}
```


Para obter mais informações, consulte Perfil do [AWS Fargate - Criação de um perfil do Fargate no Guia](#) do usuário da Amazon. EKS

Exemplo 2: Criar perfil EKS Fargate para um seletor com namespace e rótulos

O `create-fargate-profile` exemplo a seguir cria um Perfil EKS Fargate para um seletor com um namespace e rótulos.

```
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"}}]'
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte Perfil do [AWS Fargate - Criação de um perfil do Fargate no Guia](#) do usuário da Amazon. EKS

Exemplo 3: Crie um perfil EKS Fargate para um seletor com um namespace e rótulos, além IDs de sub-redes nas quais iniciar um pod.

O `create-fargate-profile` exemplo a seguir cria o Perfil EKS Fargate para um seletor com um namespace e rótulos, além IDs de sub-redes nas quais iniciar um 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"]
```

Saída:

```
{  
  "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": {}  
  }  
}
```

```
}
}
```

Para obter mais informações, consulte Perfil do [AWS Fargate - Criação de um perfil do Fargate no Guia](#) do usuário da Amazon. EKS

Exemplo 4: Crie o perfil EKS Fargate para um seletor com vários namespaces e rótulos, além IDs de sub-redes para iniciar um pod

O `create-fargate-profile` exemplo a seguir cria um Perfil EKS Fargate para um seletor com vários namespaces e rótulos, além IDs de sub-redes nas quais iniciar um 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"]
```

Saída:

```
{
  "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"
}
}
}

```

Para obter mais informações, consulte Perfil do [AWS Fargate - Criação de um perfil do Fargate no Guia](#) do usuário da Amazon. EKS

Exemplo 5: Crie um perfil EKS Fargate com um seletor curinga para namespaces e rótulos, além de sub-redes para iniciar IDs um pod

O `create-fargate-profile` exemplo a seguir cria um Perfil EKS Fargate para um seletor com vários namespaces e rótulos, além IDs de sub-redes nas quais iniciar um 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"}'

```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte Perfil do [AWS Fargate - Criação de um perfil do Fargate no Guia](#) do usuário da Amazon. EKS

- Para API obter detalhes, consulte [CreateFargateProfile](#) na Referência de AWS CLI Comandos.

create-nodegroup

O código de exemplo a seguir mostra como usar create-nodegroup.

AWS CLI

Exemplo 1: Cria um grupo de nós gerenciados para um EKS cluster da Amazon

O `create-nodegroup` exemplo a seguir cria um grupo de nós gerenciados para um EKS cluster da 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
```

Saída:

```
{
  "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": {}
  }
}

```

Para obter mais informações, consulte [Criação de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

Exemplo 2: Cria um grupo de nós gerenciados para um EKS cluster da Amazon com tipos de instância e tamanho de disco personalizados

O `create-nodegroup` exemplo a seguir cria um grupo de nós gerenciados para um EKS cluster da Amazon com tipos de instância e tamanho de disco personalizados.

```

aws eks create-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --node-role arn:aws:iam::111122223333:role/role-name \
  --
subnets "subnet-0e2907431c9988b72" "subnet-04ad87f71c6e5ab4d" "subnet-09d912bb63ef21b9a" \
  --scaling-config minSize=1,maxSize=3,desiredSize=1 \
  --capacity-type ON_DEMAND \
  --instance-types 'm5.large' \
  --disk-size 50 \
  --region us-east-2

```

Saída:

```

{
  "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": {}
}
```

Para obter mais informações, consulte [Criação de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

Exemplo 3: Cria um grupo de nós gerenciados para um EKS cluster da Amazon com tipos de instância, tamanho de disco, tipo ami, tipo de capacidade, configuração de atualização, rótulos, contaminações e tags personalizados.

O `create-nodegroup` exemplo a seguir cria um grupo de nós gerenciados para um EKS cluster da Amazon com tipos de instância, tamanho de disco, tipo ami, tipo de capacidade, configuração de atualização, rótulos, contaminações e tags personalizados.


```
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"}'
```

Saída:

```
{
  "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"
    }
}
}
```

Para obter mais informações, consulte [Criação de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

- Para API obter detalhes, consulte [CreateNodegroup](#) na Referência de AWS CLI Comandos.

delete-addon

O código de exemplo a seguir mostra como usar `delete-addon`.

AWS CLI

Exemplo 1. Para excluir um EKS complemento da Amazon, mas preservar o software complementar no cluster EKS

O comando de `delete-addon` exemplo a seguir exclui um EKS complemento da Amazon, mas preserva o software complementar no cluster. EKS

```
aws eks delete-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --preserve
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Excluindo um complemento na Amazon. EKS](#)

Exemplo 2. Para excluir um EKS complemento da Amazon e também excluir o software complementar do cluster EKS

O comando de `delete-addon` exemplo a seguir exclui um EKS complemento da Amazon e também exclui o software complementar do cluster. EKS

```
aws eks delete-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --preserve
```

```
--cluster-name my-eks-cluster \  
--addon-name my-eks-addon
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Excluindo um complemento na Amazon. EKS](#)

- Para API obter detalhes, consulte [DeleteAddon](#) na Referência de AWS CLI Comandos.

delete-cluster

O código de exemplo a seguir mostra como usar `delete-cluster`.

AWS CLI

Excluir um plano de controle de EKS cluster da Amazon

O `delete-cluster` exemplo a seguir exclui um plano de controle de EKS cluster da Amazon.

```
aws eks delete-cluster \  
--name my-eks-cluster
```

Saída:

```
{
  "cluster": {
    "name": "my-eks-cluster",
    "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster",
    "createdAt": "2024-03-14T11:31:44.348000-04:00",
    "version": "1.27",
    "endpoint": "https://DALSJ343KE23J3RN45653DSKJTT647TYD.yl4.us-
east-2.eks.amazonaws.com",
    "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-
ServiceRole-zMF6CBakwwbW",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-0fb75d2d8401716e7",
        "subnet-02184492f67a3d0f9",
        "subnet-04098063527aab776",
        "subnet-0e2907431c9988b72",
        "subnet-04ad87f71c6e5ab4d",
        "subnet-09d912bb63ef21b9a"
      ],
      "securityGroupIds": [
        "sg-0c1327f6270afbb36"
      ],
      "clusterSecurityGroupId": "sg-01c84d09d70f39a7f",
      "vpcId": "vpc-0012b8e1cc0abb17d",
      "endpointPublicAccess": true,
      "endpointPrivateAccess": true,
      "publicAccessCidrs": [
        "0.0.0.0/0"
      ]
    },
    "kubernetesNetworkConfig": {
      "serviceIpv4Cidr": "10.100.0.0/16",
      "ipFamily": "ipv4"
    },
    "logging": {
      "clusterLogging": [
        {
          "types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ]
        }
      ]
    }
  }
}
```

```

        ],
        "enabled": true
    }
]
},
"identity": {
    "oidc": {
        "issuer": "https://oidc.eks.us-east-2.amazonaws.com/id/
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"
}
}
}

```

Para obter mais informações, consulte [Excluir um EKS cluster da Amazon](#) no Guia do EKS usuário da Amazon.

- Para API obter detalhes, consulte [DeleteCluster](#) na Referência de AWS CLI Comandos.

delete-fargate-profile

O código de exemplo a seguir mostra como usar delete-fargate-profile.

AWS CLI

Exemplo 1: Criar perfil EKS Fargate para um seletor com um namespace

O delete-fargate-profile exemplo a seguir cria um Perfil EKS Fargate para um seletor com um namespace.

```
aws eks delete-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --fargate-profile-name my-fargate-profile
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Perfil do AWS Fargate - Excluindo um Fargate](#) no Guia do usuário da Amazon. EKS

- Para API obter detalhes, consulte [DeleteFargateProfile](#) na Referência de AWS CLI Comandos.

delete-nodegroup

O código de exemplo a seguir mostra como usar delete-nodegroup.

AWS CLI

Exemplo 1: Excluir um grupo de nós gerenciados para um EKS cluster da Amazon

O delete-nodegroup exemplo a seguir exclui um grupo de nós gerenciados para um EKS cluster da Amazon.

```
aws eks delete-nodegroup \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup
```

Saída:

```
{  
  "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"
    }
  }
}
```

- Para API obter detalhes, consulte [DeleteNodegroup](#) na Referência de AWS CLI Comandos.

deregister-cluster

O código de exemplo a seguir mostra como usar `deregister-cluster`.

AWS CLI

Para cancelar o registro de um cluster conectado para removê-lo do plano de controle da Amazon EKS

O `deregister-cluster` exemplo a seguir cancela o registro de um cluster conectado para removê-lo do plano de controle da AmazonEKS.

```
aws eks deregister-cluster \  
  --name my-eks-anywhere-cluster
```

Saída:

```
{  
  "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/AmazonEKSConectorAgentRole"  
    }  
  }  
}
```

Para obter mais informações, consulte [Cancelamento do registro de um cluster](#) no Guia do usuário da Amazon EKS.

- Para API obter detalhes, consulte [DeregisterCluster](#) na Referência de AWS CLI Comandos.

describe-addon-configuration

O código de exemplo a seguir mostra como usar `describe-addon-configuration`.

AWS CLI

Exemplo 1: Opções de configuração disponíveis ao criar ou atualizar o Amazon vpc-cni AddOns

O `describe-addon-configuration` exemplo a seguir retorna todo o esquema de configuração disponível que você usa quando um complemento é criado ou atualizado para o complemento vpc-cni com a respectiva versão.

```
aws eks describe-addon-configuration \  
  --name vpc-cni
```

```
--addon-name vpc-cni \
--addon-version v1.15.1-eksbuild.1
```

Saída:

```
{
  "addonName": "vpc-cni",
  "addonVersion": "v1.15.1-eksbuild.1",
  "configurationSchema": "{\ref\": \"#/definitions/VpcCni\", \"schema\": \"http://
json-schema.org/draft-06/schema#\", \"definitions\": {\Affinity\": {\type\":
[\"object\", \"null\"]}, \"EniConfig\": {\additionalProperties\": false, \"properties
\": {\create\": {\type\": \"boolean\"}, \"region\": {\type\": \"string\"}, \"subnets\":
{\additionalProperties\": {\additionalProperties\": false, \"properties\": {\id\":
{\type\": \"string\"}, \"securityGroups\": {\items\": {\type\": \"string\"}, \"type\":
\"array\"}}, \"required\": [\"id\"], \"type\": \"object\"}, \"minProperties\": 1, \"type
\": \"object\"}}, \"required\": [\"create\", \"region\", \"subnets\"], \"type\": \"object
\"}, \"Env\": {\additionalProperties\": false, \"properties\": {\ADDITIONAL_ENI_TAGS
\": {\type\": \"string\"}, \"ANNOTATE_POD_IP\": {\format\": \"boolean\", \"type\":
\"string\"}, \"AWS_EC2_ENDPOINT\": {\type\": \"string\"}, \"AWS_EXTERNAL_SERVICE_CIDRS
\": {\type\": \"string\"}, \"AWS_MANAGE_ENIS_NON_SCHEDULABLE\": {\format\": \"boolean
\", \"type\": \"string\"}, \"AWS_VPC_CNI_NODE_PORT_SUPPORT\": {\format\": \"boolean
\", \"type\": \"string\"}, \"AWS_VPC_ENI_MTU\": {\format\": \"integer\", \"type\":
\"string\"}, \"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\": {\format\": \"boolean\", \"type
\": \"string\"}, \"AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS\": {\type\": \"string\"},
\"AWS_VPC_K8S_CNI_EXTERNALSNAT\": {\format\": \"boolean\", \"type\": \"string\"},
\"AWS_VPC_K8S_CNI_LOGLEVEL\": {\type\": \"string\"}, \"AWS_VPC_K8S_CNI_LOG_FILE\":
{\type\": \"string\"}, \"AWS_VPC_K8S_CNI_RANDOMIZESNAT\": {\type\": \"string\"},
\"AWS_VPC_K8S_CNI_VETHPREFIX\": {\type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_FILE
\": {\type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\": {\type\": \"string
\"}, \"CLUSTER_ENDPOINT\": {\type\": \"string\"}, \"DISABLE_INTROSPECTION\":
{\format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_LEAKED_ENI_CLEANUP\":
{\format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_METRICS\": {\format
\": \"boolean\", \"type\": \"string\"}, \"DISABLE_NETWORK_RESOURCE_PROVISIONING
\": {\format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_POD_V6\": {\format
\": \"boolean\", \"type\": \"string\"}, \"ENABLE_BANDWIDTH_PLUGIN\": {\format\":
\"boolean\", \"type\": \"string\"}, \"ENABLE_POD_ENI\": {\format\": \"boolean\",
\"type\": \"string\"}, \"ENABLE_PREFIX_DELEGATION\": {\format\": \"boolean\",
\"type\": \"string\"}, \"ENABLE_V4_EGRESS\": {\format\": \"boolean\", \"type\":
\"string\"}, \"ENABLE_V6_EGRESS\": {\format\": \"boolean\", \"type\": \"string\"},
\"ENI_CONFIG_ANNOTATION_DEF\": {\type\": \"string\"}, \"ENI_CONFIG_LABEL_DEF\":
{\type\": \"string\"}, \"INTROSPECTION_BIND_ADDRESS\": {\type\": \"string\"},
\"IP_COOLDOWN_PERIOD\": {\format\": \"integer\", \"type\": \"string\"}, \"MAX_ENI
\": {\format\": \"integer\", \"type\": \"string\"}, \"MINIMUM_IP_TARGET\": {\format
```

```

\":"integer","\type\":"string"},\POD_SECURITY_GROUP_ENFORCING_MODE\":"
{"type\":"string"},\WARM_ENI_TARGET\":"{"format\":"integer","\type\":"
string"},\WARM_IP_TARGET\":"{"format\":"integer","\type\":"string"},
\WARM_PREFIX_TARGET\":"{"format\":"integer","\type\":"string"}},\title
\":"Env","\type\":"object"},\Init\":"{"additionalProperties\":"false,
\properties\":"{"env\":"{"$ref\":"#/definitions/InitEnv"}},\title\":"Init
","\type\":"object"},\InitEnv\":"{"additionalProperties\":"false,\properties
\":"{"DISABLE_TCP_EARLY_DEMUX\":"{"format\":"boolean","\type\":"string"},
\ENABLE_V6_EGRESS\":"{"format\":"boolean","\type\":"string"}},\title\":"
InitEnv","\type\":"object"},\Limits\":"{"additionalProperties\":"false,
\properties\":"{"cpu\":"{"type\":"string"},\memory\":"{"type\":"string"}},
\title\":"Limits","\type\":"object"},\NodeAgent\":"{"additionalProperties
\":"false,\properties\":"{"enableCloudWatchLogs\":"{"format\":"boolean",
\type\":"string"},\enablePolicyEventLogs\":"{"format\":"boolean","\type\":"
string"},\healthProbeBindAddr\":"{"format\":"integer","\type\":"string
"},\metricsBindAddr\":"{"format\":"integer","\type\":"string"}},\title\":"
NodeAgent","\type\":"object"},\Resources\":"{"additionalProperties\":"false,
\properties\":"{"limits\":"{"$ref\":"#/definitions/Limits"},\requests\":"
{"$ref\":"#/definitions/Limits"}},\title\":"Resources","\type\":"object
"},\Tolerations\":"{"additionalProperties\":"false,\items\":"{"type\":"object
"},\type\":"array"},\VpcCni\":"{"additionalProperties\":"false,\properties
\":"{"affinity\":"{"$ref\":"#/definitions/Affinity"},\enableNetworkPolicy\":"
{"format\":"boolean","\type\":"string"},\enableWindowsIpam\":"{"format\":"
boolean","\type\":"string"},\eniConfig\":"{"$ref\":"#/definitions/EniConfig
"},\env\":"{"$ref\":"#/definitions/Env"},\init\":"{"$ref\":"#/definitions/Init
"},\livenessProbeTimeoutSeconds\":"{"type\":"integer"},\nodeAgent\":"{"$ref\":"
#/definitions/NodeAgent"},\readinessProbeTimeoutSeconds\":"{"type\":"integer
"},\resources\":"{"$ref\":"#/definitions/Resources"},\tolerations\":"{"$ref
\":"#/definitions/Tolerations"}},\title\":"VpcCni","\type\":"object"}},
\description\":"vpc-cni"}"
}

```

Exemplo 2: Opções de configuração disponíveis ao criar ou atualizar coredns da Amazon AddOns

O `describe-addon-configuration` exemplo a seguir retorna todo o esquema de configuração disponível que você usa quando um complemento é criado ou atualizado para o complemento `coredns` com a respectiva versão.

```

aws eks describe-addon-configuration \
  --addon-name coredns \
  --addon-version v1.8.7-eksbuild.4

```

Saída:

```
{
  "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\"}}}"
}
```

Para obter mais informações, consulte [Criação ou atualização de um arquivo kubeconfig para um EKS cluster da Amazon na Amazon](#). EKS

- Para API obter detalhes, consulte [DescribeAddonConfiguration](#) na Referência de AWS CLI Comandos.

describe-addon-versions

O código de exemplo a seguir mostra como usar `describe-addon-versions`.

AWS CLI

Exemplo 1: Listar todos os complementos disponíveis para EKS o Cluster

O `describe-addon-versions` exemplo a seguir lista todos os AWS complementos disponíveis.

```
aws eks describe-addon-versions \
  --query 'sort_by(addons &owner)[].{publisher: publisher, owner: owner,
  addonName: addonName, type: type}' \
  --output table
```

Saída:

```
-----
```

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 |
+-----+
+-----+

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criação de um complemento no Guia EKS do usuário da Amazon](#).

Exemplo 2: Listar todos os complementos disponíveis para a versão especificada do Kubernetes com suporte para EKS

O `describe-addon-versions` exemplo a seguir lista todos os complementos disponíveis para a versão especificada do Kubernetes com suporte. 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
```

Saída:

```
-----
|                                     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                       |                  |
+-----+-----+
+-----+-----+

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criação de um complemento no Guia EKS do usuário da Amazon](#).

Exemplo 3: Listar todas as versões de complementos vpc-cni disponíveis para a versão especificada do Kubernetes compatível com EKS

O `describe-addon-versions` exemplo a seguir lista todas as versões de complementos vpc-cni disponíveis para a versão especificada do Kubernetes com suporte. EKS

```
aws eks describe-addon-versions \
  --kubernetes-version=1.26 \
  --addon-name=vpc-cni \
  --query='addons[].addonVersions[].addonVersion'
```

Saída:

```
[
  "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"
]
```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Criação de um complemento no Guia EKS do usuário da Amazon](#).

- Para API obter detalhes, consulte [DescribeAddonVersions](#) na Referência de AWS CLI Comandos.

describe-addon

O código de exemplo a seguir mostra como usar `describe-addon`.

AWS CLI

Descreva o EKS complemento em execução ativa em seu cluster da Amazon EKS

O `describe-addon` exemplo a seguir executando ativamente o EKS complemento em seu EKS cluster da Amazon.

```
aws eks describe-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name vpc-cni
```

Saída:

```
{  
  "addon": {  
    "addonName": "vpc-cni",  
    "clusterName": "my-eks-cluster",  
    "status": "ACTIVE",  
    "addonVersion": "v1.16.4-eksbuild.2",  
    "health": {  
      "issues": []  
    },  
    "addonArn": "arn:aws:eks:us-east-2:111122223333:addon/my-eks-cluster/vpc-cni/0ec71efc-98dd-3203-60b0-4b939b2a5e5f",  
    "createdAt": "2024-03-14T13:18:45.417000-04:00",  
    "modifiedAt": "2024-03-14T13:18:49.557000-04:00",  
    "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-addon-vpc-cni-Role1-Yfakrq0C1UTm",  
    "tags": {  
      "eks-addon-key-3": "value-3",  
      "eks-addon-key-4": "value-4"  
    },  
    "configurationValues": "resources:\n      limits:\n        cpu: '100m'\n      nenv:\n        AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'  
  }  
}
```

- Para API obter detalhes, consulte [DescribeAddon](#) na Referência de AWS CLI Comandos.

describe-cluster

O código de exemplo a seguir mostra como usar `describe-cluster`.

AWS CLI

Descreva o EKS complemento em execução ativa em seu cluster da Amazon EKS

O `describe-cluster` exemplo a seguir executando ativamente o EKS complemento em seu EKS cluster da Amazon.

```
aws eks describe-cluster \
  --cluster-name my-eks-cluster
```

Saída:

```
{
  "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"
    }
}
}
}

```

- Para API obter detalhes, consulte [DescribeCluster](#) na Referência de AWS CLI Comandos.

describe-fargate-profile

O código de exemplo a seguir mostra como usar `describe-fargate-profile`.

AWS CLI

Descreva um perfil do Fargate

O `describe-fargate-profile` exemplo a seguir descreve um perfil do Fargate.

```

aws eks describe-fargate-profile \
  --cluster-name my-eks-cluster \
  --fargate-profile-name my-fargate-profile

```

Saída:

```

{
  "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"
    }
}
}

```

- Para API obter detalhes, consulte [DescribeFargateProfile](#) na Referência de AWS CLI Comandos.

describe-identity-provider-config

O código de exemplo a seguir mostra como usar `describe-identity-provider-config`.

AWS CLI

Descreva uma configuração de provedor de identidade associada ao seu Amazon EKS Cluster

O `describe-identity-provider-config` exemplo a seguir descreve uma configuração de provedor de identidade associada ao seu Amazon EKS Cluster.

```

aws eks describe-identity-provider-config \
  --cluster-name my-eks-cluster \
  --identity-provider-config type=oidc,name=my-identity-provider

```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Autenticar usuários para seu cluster a partir de um provedor de identidade OpenID Connect](#) no Guia do usuário da EKS Amazon.

- Para API obter detalhes, consulte [DescribeIdentityProviderConfig](#) na Referência de AWS CLI Comandos.

describe-nodegroup

O código de exemplo a seguir mostra como usar `describe-nodegroup`.

AWS CLI

Descrever um grupo de nós gerenciados para um EKS cluster da Amazon

O `describe-nodegroup` exemplo a seguir descreve um grupo de nós gerenciados para um EKS cluster da Amazon.

```
aws eks describe-nodegroup \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup
```

Saída:

```
{  
  "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": {}
}
}

```

- Para API obter detalhes, consulte [DescribeNodegroup](#) na Referência de AWS CLI Comandos.

describe-update

O código de exemplo a seguir mostra como usar `describe-update`.

AWS CLI

Exemplo 1: Para descrever uma atualização para um cluster

O `describe-update` exemplo a seguir descreve uma atualização para um cluster chamado.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Atualização de uma versão do Kubernetes EKS do cluster da Amazon no Guia](#) do usuário da Amazon EKS.

Exemplo 2: Para descrever uma atualização para um cluster

O describe-update exemplo a seguir descreve uma atualização para um cluster chamado.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id e4994991-4c0f-475a-a040-427e6da52966

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Atualização de uma versão do Kubernetes EKS do cluster da Amazon no Guia](#) do usuário da Amazon EKS.

Exemplo 3: Para descrever uma atualização para um cluster

O describe-update exemplo a seguir descreve uma atualização para um cluster chamado.

```
aws eks describe-update \  
  --name my-eks-cluster \  
  --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Atualização de uma versão do Kubernetes EKS do cluster da Amazon no Guia](#) do usuário da Amazon EKS.

- Para API obter detalhes, consulte [DescribeUpdate](#) na Referência de AWS CLI Comandos.

disassociate-identity-provider-config

O código de exemplo a seguir mostra como usar disassociate-identity-provider-config.

AWS CLI

Desassocie o provedor de identidade do seu Amazon Cluster EKS

O disassociate-identity-provider-config exemplo a seguir desassocia um provedor de identidade do seu Amazon EKS Cluster.

```
aws eks disassociate-identity-provider-config \  
  --cluster-name my-eks-cluster \  
  --identity-provider-config 'type=oidc,name=my-identity-provider'
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Autenticar usuários do seu cluster a partir de um provedor de identidade do OpenID Connect - OIDC Desassociar um provedor de identidade do seu cluster no Guia do](#) usuário da EKS Amazon.

- Para API obter detalhes, consulte [DisassociateIdentityProviderConfig](#) Referência de AWS CLI Comandos.

get-token

O código de exemplo a seguir mostra como usar get-token.

AWS CLI

Exemplo 1: Obtenha um token de autenticação para um EKS cluster da Amazon chamado `my-eks-cluster`

O get-token exemplo a seguir obtém um token de autenticação para um EKS cluster da Amazon chamado my-eks-cluster.

```
aws eks get-token \  
  --cluster-name my-eks-cluster
```

```
--cluster-name my-eks-cluster
```

Saída:

```
{
  "kind": "ExecCredential",
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "spec": {},
  "status": {
    "expirationTimestamp": "2024-04-11T20:59:56Z",
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
  }
}
```

Exemplo 2: Obtém um token de autenticação para um EKS cluster da Amazon chamado `my-eks-cluster` assumindo essa função ARN para credenciais ao assinar o token

O `get-token` exemplo a seguir obtém um token de autenticação para um EKS cluster da Amazon nomeado `my-eks-cluster` assumindo essa função ARN de credenciais ao assinar o 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
```

Saída:

```
{
  "kind": "ExecCredential",
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "spec": {},
  "status": {
    "expirationTimestamp": "2024-04-11T21:05:26Z",
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
  }
}
```

- Para API obter detalhes, consulte [GetToken](#) na Referência de AWS CLI Comandos.

list-addons

O código de exemplo a seguir mostra como usar `list-addons`.

AWS CLI

Liste todos os complementos instalados em seu EKS cluster Amazon chamado `my-eks-cluster`

O `list-addons` exemplo a seguir lista todos os complementos instalados em seu EKS cluster Amazon chamado `my-eks-cluster`.

```
aws eks list-addons \
  --cluster-name my-eks-cluster
```

Saída:

```
{
  "addons": [
    "kube-proxy",
    "vpc-cni"
  ]
}
```

- Para API obter detalhes, consulte [ListAddons](#) na Referência de AWS CLI Comandos.

list-clusters

O código de exemplo a seguir mostra como usar `list-clusters`.

AWS CLI

Para listar todos os complementos instalados em seu EKS cluster Amazon chamado `my-eks-cluster`

O `list-clusters` exemplo a seguir lista todos os complementos instalados em seu EKS cluster Amazon chamado `my-eks-cluster`.

```
aws eks list-clusters
```

Saída:

```
{
```

```
"clusters": [  
  "prod",  
  "qa",  
  "stage",  
  "my-eks-cluster"  
]  
}
```

- Para API obter detalhes, consulte [ListClusters](#) na Referência de AWS CLI Comandos.

list-fargate-profiles

O código de exemplo a seguir mostra como usar `list-fargate-profiles`.

AWS CLI

Para listar todos os perfis de fargate em seu EKS cluster da Amazon chamado `my-eks-cluster`

O `list-fargate-profiles` exemplo a seguir lista todos os perfis de fargate em seu EKS cluster da Amazon denominados `my-eks-cluster`.

```
aws eks list-fargate-profiles \  
  --cluster-name my-eks-cluster
```

Saída:

```
{  
  "fargateProfileNames": [  
    "my-fargate-profile"  
  ]  
}
```

- Para API obter detalhes, consulte [ListFargateProfiles](#) na Referência de AWS CLI Comandos.

list-identity-provider-configs

O código de exemplo a seguir mostra como usar `list-identity-provider-configs`.

AWS CLI

Listar provedores de identidade associados a um EKS cluster da Amazon

O `list-identity-provider-configs` exemplo a seguir lista o provedor de identidade associado a um EKS cluster da Amazon.

```
aws eks list-identity-provider-configs \  
  --cluster-name my-eks-cluster
```

Saída:

```
{  
  "identityProviderConfigs": [  
    {  
      "type": "oidc",  
      "name": "my-identity-provider"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Autenticar usuários para seu cluster a partir de um provedor de identidade OpenID Connect](#) no Guia do usuário da EKS Amazon.

- Para API obter detalhes, consulte [ListIdentityProviderConfigs](#) na Referência de AWS CLI Comandos.

list-nodegroups

O código de exemplo a seguir mostra como usar `list-nodegroups`.

AWS CLI

Listar todos os grupos de nós em um EKS cluster da Amazon

O `list-nodegroups` exemplo a seguir lista todos os grupos de nós em um EKS cluster da Amazon.

```
aws eks list-nodegroups \  
  --cluster-name my-eks-cluster
```

Saída:

```
{  
  "nodegroups": [  
    "my-eks-managed-node-group",  
  ]  
}
```

```

    "my-eks-nodegroup"
  ]
}

```

- Para API obter detalhes, consulte [ListNodegroups](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Exemplo 1: Para listar todas as tags de um EKS cluster da Amazon ARN

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um EKS cluster da Amazon ARN.

```

aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 2: Para listar todas as tags de um grupo Amazon EKS Node ARN

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um grupo 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
```

Saída:

```
{
  "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/ea20310-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"
  }
}
```

Exemplo 3: Para listar todas as tags em um perfil do Amazon EKS Fargate ARNe

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um perfil do 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
```

Saída:

```
{
  "tags": {
    "eks-fargate-profile-key-2": "value-2",
    "eks-fargate-profile-key-1": "value-1"
  }
}
```

Exemplo 4: Para listar todas as tags de um EKS complemento da Amazon ARN

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um EKS complemento ARN da Amazon.

```
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
```

Saída:

```
{
  "tags": {
    "eks-addon-key-2": "value-2",
    "eks-addon-key-1": "value-1"
  }
}
```

Exemplo 5: Para listar todas as tags de um provedor de EKS OIDC identidade da Amazon ARN

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um provedor de EKS OIDC identidade da 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
```

Saída:

```
{
  "tags": {
    "my-identity-provider": "test"
  }
}
```

```
}  
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-update

O código de exemplo a seguir mostra como usar `list-update`.

AWS CLI

Exemplo 1: Para listar as atualizações associadas a um nome de EKS cluster da Amazon

O `list-updates` exemplo a seguir lista todas as atualizações IDs para um nome de EKS cluster da Amazon.

```
aws eks list-updates \  
  --name my-eks-cluster
```

Saída:

```
{  
  "updateIds": [  
    "5f78d14e-c57b-4857-a3e4-cf664ae20949",  
    "760e5a3f-adad-48c7-88d3-7ac283c09c26",  
    "cd4ec863-bc55-47d5-a377-3971502f529b",  
    "f12657ce-e869-4f17-b158-a82ab8b7d937"  
  ]  
}
```

Exemplo 2: Para listar todas as atualizações IDs de um grupo Amazon EKS Node

O `list-updates` exemplo a seguir lista todas as atualizações IDs de um grupo Amazon EKS Node.

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --nodegroup-name my-eks-managed-node-group
```

Saída:

```
{
  "updateIds": [
    "8c6c1bef-61fe-42ac-a242-89412387b8e7"
  ]
}
```

Exemplo 3: Para listar todas as atualizações IDs em um EKS complemento da Amazon

O `list-updates` exemplo a seguir lista todas as atualizações IDs de um EKS complemento da Amazon.

```
aws eks list-updates \
  --name my-eks-cluster \
  --addon-name vpc-cni
```

Saída:

```
{
  "updateIds": [
    "9cdba8d4-79fb-3c83-afe8-00b508d33268"
  ]
}
```

- Para API obter detalhes, consulte [ListUpdate](#) na Referência de AWS CLI Comandos.

list-updates

O código de exemplo a seguir mostra como usar `list-updates`.

AWS CLI

Para listar as atualizações de um cluster

Este exemplo de comando lista as atualizações atuais de um cluster chamado `example` na sua região padrão.

Comando:

```
aws eks list-updates --name example
```

Saída:

```
{
  "updateIds": [
    "10bddb13-a71b-425a-b0a6-71cd03e59161"
  ]
}
```

- Para API obter detalhes, consulte [ListUpdates](#) na Referência de AWS CLI Comandos.

register-cluster

O código de exemplo a seguir mostra como usar `register-cluster`.

AWS CLI

Exemplo 1: registrar um cluster externo EKS _ ANYWHERE Kubernetes na Amazon EKS

O `register-cluster` exemplo a seguir registra um ANYWHERE cluster externo EKS _ Kubernetes na Amazon. EKS

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::11112223333:role/AmazonEKSCoordinatorAgentRole,provider=EKS_ANYWHERE'
```

Saída:

```
{
  "cluster": {
    "name": "my-eks-anywhere-cluster",
    "arn": "arn:aws:eks:us-east-2:11112223333: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::11112223333:role/AmazonEKSCoordinatorAgentRole"
    }
  }
}
```

```
}
```

Para obter mais informações, consulte [Conectando um cluster externo](#) no Guia EKS do usuário da Amazon.

Exemplo 2: Registre qualquer cluster externo do Kubernetes na Amazon EKS

O `register-cluster` exemplo a seguir registra um ANYWHERE cluster externo EKS _ Kubernetes na Amazon. EKS

```
aws eks register-cluster \  
  --name my-eks-anywhere-cluster \  
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/  
AmazonEKSCollectorAgentRole,provider=OTHER'
```

Saída:

```
{  
  "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/AmazonEKSCollectorAgentRole"  
    }  
  }  
}
```

Para obter mais informações, consulte [Conectando um cluster externo](#) no Guia EKS do usuário da Amazon.

- Para API obter detalhes, consulte [RegisterCluster](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Exemplo 1: Para adicionar as tags especificadas a um EKS cluster da Amazon

O `tag-resource` exemplo a seguir adiciona as tags especificadas a um EKS cluster da Amazon.

```
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'
```

Este comando não produz saída.

Exemplo 2: Para adicionar as tags especificadas a um grupo Amazon EKS Node

O `tag-resource` exemplo a seguir adiciona as tags especificadas a um grupo 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'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Exemplo 1: Para excluir as tags especificadas de um cluster da Amazon EKS

O `untag-resource` exemplo a seguir exclui as tags especificadas de um EKS cluster da Amazon.

```
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"
```

Este comando não produz saída.

Exemplo 2: Para excluir as tags especificadas de um grupo Amazon EKS Node

O `untag-resource` exemplo a seguir exclui as tags especificadas de um grupo 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"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-addon

O código de exemplo a seguir mostra como usar `update-addon`.

AWS CLI

Exemplo 1. Para atualizar um EKS complemento da Amazon com a função de conta de serviço ARN

O comando de `update-addon` exemplo a seguir atualiza um EKS complemento da Amazon com a função ARN de conta de serviço.

```
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
```

Saída:

```
{  
  "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": []
}
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Atualizando um complemento no Guia EKS](#) do usuário da Amazon.

Exemplo 2. Para atualizar um EKS complemento da Amazon com uma versão específica

O comando de update-addon exemplo a seguir atualiza um EKS complemento da Amazon com uma versão específica do complemento.

```

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

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Atualizando um complemento no Guia EKS](#) do usuário da Amazon.

Exemplo 3. Para atualizar um EKS complemento da Amazon com valores de configuração personalizados e resolver detalhes de conflitos

O comando de update-addon exemplo a seguir atualiza um EKS complemento da Amazon com valores de configuração personalizados e detalhes de resolução de conflitos.

```

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

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Atualizando um complemento no Guia EKS](#) do usuário da Amazon.

Exemplo 4. Para atualizar um EKS complemento da Amazon com um arquivo de valores JSON de configuração personalizados

O comando de update-addon exemplo a seguir atualiza um EKS complemento da Amazon com valores de JSON configuração personalizados e detalhes de resolução de conflitos.

```

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

```

Conteúdo de configuration-values.json:

```

{
  "resources": {
    "limits": {
      "cpu": "100m"
    },
    "requests": {
      "cpu": "50m"
    }
  },
  "env": {

```

```

    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
  }
}

```

Saída:

```

{
  "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
      \"memory\": \"128Mi\"\n    }\n  },\n  \"env\": {\n    \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR
      \"\n    }\n}"
      }
    ],
    "createdAt": "2024-04-12T16:22:55.519000-04:00",
    "errors": []
  }
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Atualizando um complemento no Guia EKS](#) do usuário da Amazon.

Exemplo 5. Para atualizar um EKS complemento da Amazon com um arquivo de valores YAML de configuração personalizados

O comando de `update-addon` exemplo a seguir atualiza um EKS complemento da Amazon com valores de YAML configuração personalizados e detalhes de resolução de conflitos.

```
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
```

Conteúdo de `configuration-values.yaml`:

```
resources:
  limits:
    cpu: '100m'
  requests:
    cpu: '50m'
env:
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

Saída:

```
{
  "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'\nrequests:\n        cpu: '50m'\nenv:\n    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
    }
  ],
  "createdAt": "2024-04-12T16:25:07.212000-04:00",
  "errors": []
}
}

```

Para obter mais informações, consulte [Gerenciando EKS complementos da Amazon - Atualizando um complemento no Guia EKS](#) do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateAddon](#) na Referência de AWS CLI Comandos.

update-cluster-config

O código de exemplo a seguir mostra como usar `update-cluster-config`.

AWS CLI

Para atualizar o acesso ao endpoint do cluster

Este exemplo de comando atualiza um cluster para desativar o acesso público ao endpoint e habilitar o acesso ao endpoint privado.

Comando:

```
aws eks update-cluster-config --name example \
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

Saída:

```

{
  "update": {
    "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",
    "status": "InProgress",
    "type": "EndpointAccessUpdate",
    "params": [
      {

```



```

        "type": "EndpointPublicAccess",
        "value": "false"
      },
      {
        "type": "EndpointPrivateAccess",
        "value": "true"
      }
    ],
    "createdAt": 1565806986.506,
    "errors": []
  }
}

```

Para habilitar o registro em um cluster

Este exemplo de comando habilita todos os tipos de registro do plano de controle de cluster para um cluster chamado `example`.

Comando:

```

aws eks update-cluster-config --name example \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'

```

Saída:

```

{
  "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": []
  }
}

```

- Para API obter detalhes, consulte [UpdateClusterConfig](#) na Referência de AWS CLI Comandos.

update-cluster-version

O código de exemplo a seguir mostra como usar `update-cluster-version`.

AWS CLI

Para atualizar um EKS cluster da Amazon chamado `my-eks-cluster` para a versão especificada do Kubernetes

O `update-cluster-version` exemplo a seguir atualiza um EKS cluster da Amazon para a versão especificada do Kubernetes.

```
aws eks update-cluster-version \  
  --name my-eks-cluster \  
  --kubernetes-version 1.27
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Atualização de uma versão do Kubernetes EKS do cluster da Amazon no Guia](#) do usuário da Amazon EKS.

- Para API obter detalhes, consulte [UpdateClusterVersion](#) na Referência de AWS CLI Comandos.

update-kubeconfig

O código de exemplo a seguir mostra como usar update-kubeconfig.

AWS CLI

Exemplo 1: configura seu kubectl criando ou atualizando o kubeconfig para que você possa se conectar a um cluster da Amazon chamado `my-eks-cluster`

O update-kubeconfig exemplo a seguir configura seu kubectl criando ou atualizando o kubeconfig para que você possa se conectar a um cluster da Amazon chamado. EKS my-eks-cluster

```
aws eks update-kubeconfig \  
  --name my-eks-cluster
```

Saída:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/  
xxx/.kube/config
```

Para obter mais informações, consulte [Criação ou atualização de um arquivo kubeconfig para um EKS cluster da Amazon](#) no Guia do usuário da Amazon EKS.

Exemplo 2: configura seu kubectl criando ou atualizando o kubeconfig (com a opção role-arn para assumir uma função para autenticação de cluster) para que você possa se conectar a um cluster da Amazon chamado `my-eks-cluster`

O update-kubeconfig exemplo a seguir configura seu kubectl criando ou atualizando o kubeconfig (com a opção role-arn para assumir uma função para autenticação de cluster) para que você possa se conectar a um cluster da Amazon chamado. 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
```

Saída:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/xxx/.kube/config
```

Para obter mais informações, consulte [Criação ou atualização de um arquivo kubeconfig para um EKS cluster da Amazon](#) no Guia do usuário da Amazon EKS.

Exemplo 3: configura seu kubectl criando ou atualizando o kubeconfig (com a opção `role-arn` para assumir uma função de autenticação de cluster junto com o alias de cluster e o alias de usuário personalizados) para que você possa se conectar a um cluster da Amazon chamado `EKS my-eks-cluster`

O `update-kubeconfig` exemplo a seguir configura seu kubectl criando ou atualizando o kubeconfig (com a opção `role-arn` para assumir uma função de autenticação de cluster junto com o alias de cluster e o alias de usuário personalizados) para que você possa se conectar a um cluster da Amazon chamado. `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
```

Saída:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

Para obter mais informações, consulte [Criação ou atualização de um arquivo kubeconfig para um EKS cluster da Amazon](#) no Guia do usuário da Amazon EKS.

Exemplo 4: Imprima entradas do arquivo kubeconfig para análise e configure seu kubectl para que você possa se conectar a um cluster da Amazon chamado `EKS my-eks-cluster`

O `update-kubeconfig` exemplo a seguir configura seu kubectl criando ou atualizando o kubeconfig (com a opção `role-arn` para assumir uma função de autenticação de cluster junto com o alias de cluster e o alias de usuário personalizados) para que você possa se conectar a um cluster da Amazon chamado. `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
```

Saída:

```
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
```

Para obter mais informações, consulte [Criação ou atualização de um arquivo kubeconfig para um EKS cluster da Amazon](#) no Guia do usuário da Amazon EKS.

- Para API obter detalhes, consulte [UpdateKubeconfig](#) na Referência de AWS CLI Comandos.

update-nodegroup-config

O código de exemplo a seguir mostra como usar `update-nodegroup-config`.

AWS CLI

Exemplo 1: Atualize um grupo de nós gerenciados para adicionar novos rótulos e contaminar o nó de EKS trabalho de um cluster da Amazon EKS

O `update-nodegroup-config` exemplo a seguir atualiza um grupo de nós gerenciados para adicionar novos rótulos e manchar o nó de EKS trabalho de um EKS cluster da Amazon.

```
aws eks update-nodegroup-config \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --labels 'add0rUpdateLabels={my-eks-nodegroup-label-1=value-1,my-eks-nodegroup-label-2=value-2}' \  
  --taints 'add0rUpdateTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Atualização de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

Exemplo 2: Atualize um grupo de nós gerenciados para remover rótulos e manchas do nó de EKS trabalho de um cluster da Amazon EKS

O `update-nodegroup-config` exemplo a seguir atualiza um grupo de nós gerenciados para remover rótulos e manchas do nó de EKS trabalho de um EKS cluster da 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}]'
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Atualização de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

Exemplo 3: Atualizar um grupo de nós gerenciados para remover e adicionar rótulos e manchas ao nó de EKS trabalho de um cluster da Amazon EKS

O `update-nodegroup-config` exemplo a seguir atualiza um grupo de nós gerenciados para remover e adicionar rótulos e manchas ao nó de EKS trabalho de um EKS cluster da 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}]'
```

Saída:

```
{
  "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": []
}
}
```

Para obter mais informações, consulte [Atualização de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

Exemplo 4: Atualizar um grupo de nós gerenciados para atualizar scaling-config e update-config para o nó de trabalho EKS de um cluster da Amazon EKS

O update-nodegroup-config exemplo a seguir atualiza um grupo de nós gerenciados para atualizar scaling-config e update-config para o EKS nó de trabalho de um cluster da 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
```

Saída:

```
{
  "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": []
}
}
```

Para obter mais informações, consulte [Atualização de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateNodegroupConfig](#) na Referência de AWS CLI Comandos.

update-nodegroup-version

O código de exemplo a seguir mostra como usar `update-nodegroup-version`.

AWS CLI

Exemplo 1: atualizar a versão do Kubernetes ou a versão de um grupo de AMI nós gerenciados pela Amazon EKS

O `update-nodegroup-version` exemplo a seguir atualiza a versão do Kubernetes ou AMI a versão de um grupo de nós EKS gerenciados pela Amazon para a versão mais recente disponível para seu cluster Kubernetes.

```
aws eks update-nodegroup-version \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --no-force
```

Saída:

```
{
  "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": []
  }
}

```

Para obter mais informações, consulte [Atualização de um grupo de nós gerenciados](#) no Guia EKS do usuário da Amazon.

Exemplo 2: Atualize a versão do Kubernetes ou a versão de um grupo de AMI nós gerenciados pela Amazon EKS

O `update-nodegroup-version` exemplo a seguir atualiza a versão do Kubernetes ou a versão de um grupo de nós EKS gerenciados pela Amazon para a AMI versão de lançamento especificadaAMI.

```

aws eks update-nodegroup-version \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup \
  --kubernetes-version '1.26' \
  --release-version '1.26.12-20240307' \
  --no-force

```

Saída:

```

{
  "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": []
  }
}
```

Para obter mais informações, consulte *Atualização de um grupo de nós gerenciados* - < <https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html> > no Guia do usuário da Amazon. EKS

- Para API obter detalhes, consulte [UpdateNodegroupVersion](#) na Referência de AWS CLI Comandos.

Exemplos do Elastic Beanstalk usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Elastic Beanstalk.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

abort-environment-update

O código de exemplo a seguir mostra como usar `abort-environment-update`.

AWS CLI

Para abortar uma implantação

O comando a seguir aborta a implantação de uma versão do aplicativo em execução em um ambiente chamado `my-env`:

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- Para API obter detalhes, consulte [AbortEnvironmentUpdate](#) na Referência de AWS CLI Comandos.

check-dns-availability

O código de exemplo a seguir mostra como usar `check-dns-availability`.

AWS CLI

Para verificar a disponibilidade de um CNAME

O comando a seguir verifica a disponibilidade do subdomínio: `my-cname.elasticbeanstalk.com`

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

Saída:

```
{
  "Available": true,
  "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- Para API obter detalhes, consulte [CheckDnsAvailability](#) na Referência de AWS CLI Comandos.

create-application-version

O código de exemplo a seguir mostra como usar `create-application-version`.

AWS CLI

Para criar uma nova versão do aplicativo

O comando a seguir cria uma nova versão, “v1”, de um aplicativo chamado “: 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
```

O aplicativo será criado automaticamente se ainda não existir, devido à auto-create-application opção. O pacote de origem é um arquivo .war armazenado em um bucket s3 chamado “my-bucket” que contém o aplicativo de amostra Apache Tomcat.

Saída:

```
{
  "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"
    }
  }
}
```

- Para API obter detalhes, consulte [CreateApplicationVersion](#) na Referência de AWS CLI Comandos.

create-application

O código de exemplo a seguir mostra como usar create-application.

AWS CLI

Para criar um novo aplicativo

O comando a seguir cria um novo aplicativo chamado "MyApp":

```
aws elasticbeanstalk create-application --application-name MyApp --description "my
application"
```

O `create-application` comando configura somente o nome e a descrição do aplicativo. Para carregar o código-fonte do aplicativo, crie uma versão inicial do aplicativo usando `create-application-version`. `create-application-version` também tem uma `auto-create-application` opção que permite criar o aplicativo e a versão do aplicativo em uma única etapa.

Saída:

```
{
  "Application": {
    "ApplicationName": "MyApp",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-02-12T18:32:21.181Z",
    "Description": "my application",
    "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- Para API obter detalhes, consulte [CreateApplication](#) na Referência de AWS CLI Comandos.

create-configuration-template

O código de exemplo a seguir mostra como usar `create-configuration-template`.

AWS CLI

Para criar um modelo de configuração

O comando a seguir cria um modelo `my-app-v1` de configuração com o nome das configurações aplicadas a um ambiente com o `ide-rpqsewtp2j`:

```
aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

Saída:

```
{
  "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"
```

```
}
```

- Para API obter detalhes, consulte [CreateConfigurationTemplate](#) na Referência de AWS CLI Comandos.

create-environment

O código de exemplo a seguir mostra como usar `create-environment`.

AWS CLI

Para criar um novo ambiente para um aplicativo

O comando a seguir cria um novo ambiente para a versão “v1” de um aplicativo java chamado “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"
```

Saída:

```
{
  "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 é o rótulo de uma versão do aplicativo carregada anteriormente com `create-application-version`.

Para especificar um JSON arquivo para definir as opções de configuração do ambiente

O `create-environment` comando a seguir especifica que um JSON arquivo com o nome `myoptions.json` deve ser usado para substituir valores obtidos da pilha de soluções ou do modelo de configuração:

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-name sampleapp --option-settings file://myoptions.json
```

`myoptions.json` é um JSON objeto que define várias configurações:

```
[
  {
    "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"
  }
]
```

Para obter mais informações, consulte [Valores das opções](#) no Guia do desenvolvedor do AWS Elastic Beanstalk.

- Para API obter detalhes, consulte [CreateEnvironment](#) na Referência de AWS CLI Comandos.

create-storage-location

O código de exemplo a seguir mostra como usar `create-storage-location`.

AWS CLI

Para criar um local de armazenamento

O comando a seguir cria um local de armazenamento no Amazon S3:

```
aws elasticbeanstalk create-storage-location
```

Saída:

```
{
  "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- Para API obter detalhes, consulte [CreateStorageLocation](#) na Referência de AWS CLI Comandos.

delete-application-version

O código de exemplo a seguir mostra como usar delete-application-version.

AWS CLI

Para excluir uma versão do aplicativo

O comando a seguir exclui uma versão do aplicativo com o nome 22a0-stage-150819_182129 de um aplicativo chamado my-app:

```
aws elasticbeanstalk delete-application-version --version-label 22a0-stage-150819_182129 --application-name my-app
```

- Para API obter detalhes, consulte [DeleteApplicationVersion](#) na Referência de AWS CLI Comandos.

delete-application

O código de exemplo a seguir mostra como usar delete-application.

AWS CLI

Como excluir uma aplicação

O comando a seguir exclui um aplicativo chamado `my-app`:

```
aws elasticbeanstalk delete-application --application-name my-app
```

- Para API obter detalhes, consulte [DeleteApplication](#) na Referência de AWS CLI Comandos.

delete-configuration-template

O código de exemplo a seguir mostra como usar `delete-configuration-template`.

AWS CLI

Para excluir um modelo de configuração

O comando a seguir exclui um modelo de configuração com o nome `my-template` de um aplicativo chamado `my-app`:

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --application-name my-app
```

- Para API obter detalhes, consulte [DeleteConfigurationTemplate](#) na Referência de AWS CLI Comandos.

delete-environment-configuration

O código de exemplo a seguir mostra como usar `delete-environment-configuration`.

AWS CLI

Para excluir um rascunho de configuração

O comando a seguir exclui um rascunho de configuração para um ambiente chamado `my-env`:

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --application-name my-app
```

- Para API obter detalhes, consulte [DeleteEnvironmentConfiguration](#) na Referência de AWS CLI Comandos.

describe-application-versions

O código de exemplo a seguir mostra como usar `describe-application-versions`.

AWS CLI

Para ver informações sobre uma versão do aplicativo

O comando a seguir recupera informações sobre uma versão do aplicativo chamada `v2`:

```
aws elasticbeanstalk describe-application-versions --application-name my-app --  
version-label "v2"
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeApplicationVersions](#) na Referência de AWS CLI Comandos.

describe-applications

O código de exemplo a seguir mostra como usar describe-applications.

AWS CLI

Para ver uma lista de aplicativos

O comando a seguir recupera informações sobre aplicativos na região atual:

```
aws elasticbeanstalk describe-applications
```

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [DescribeApplications](#) na Referência de AWS CLI Comandos.

describe-configuration-options

O código de exemplo a seguir mostra como usar `describe-configuration-options`.

AWS CLI

Para ver as opções de configuração de um ambiente

O comando a seguir recupera as descrições de todas as opções de configuração disponíveis para um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --
application-name my-app
```

Saída (abreviada):

```

{
  "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"
    }
]
}

```

As opções de configuração disponíveis variam de acordo com a plataforma e a versão da configuração. Para obter mais informações sobre namespaces e opções compatíveis, consulte Valores de opção no Guia do desenvolvedor do Elastic AWS Beanstalk.

- Para API obter detalhes, consulte [DescribeConfigurationOptions](#) na Referência de AWS CLI Comandos.

describe-configuration-settings

O código de exemplo a seguir mostra como usar `describe-configuration-settings`.

AWS CLI

Para visualizar as configurações de um ambiente

O comando a seguir recupera as configurações de um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --  
application-name my-app
```

Saída (abreviada):

```
{  
  "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"
      }
    ]
  }
}
```

Para obter mais informações sobre namespaces e opções compatíveis, consulte Valores de opção no Guia do desenvolvedor do Elastic AWS Beanstalk.

- Para API obter detalhes, consulte [DescribeConfigurationSettings](#) na Referência de AWS CLI Comandos.

describe-environment-health

O código de exemplo a seguir mostra como usar `describe-environment-health`.

AWS CLI

Para ver a saúde do meio ambiente

O comando a seguir recupera informações gerais de saúde de um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --
attribute-names ALL
```

Saída:

```
{
  "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": []
}
```

As informações de saúde só estão disponíveis para ambientes com relatórios de saúde aprimorados habilitados. Para obter mais informações, consulte Enhanced Health Reporting and Monitoring no Guia do Desenvolvedor do AWS Elastic Beanstalk.

- Para API obter detalhes, consulte [DescribeEnvironmentHealth](#) na Referência de AWS CLI Comandos.

describe-environment-resources

O código de exemplo a seguir mostra como usar `describe-environment-resources`.

AWS CLI

Para visualizar informações sobre os AWS recursos em seu ambiente

O comando a seguir recupera informações sobre recursos em um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [DescribeEnvironmentResources](#) na Referência de AWS CLI Comandos.

describe-environments

O código de exemplo a seguir mostra como usar `describe-environments`.

AWS CLI

Para visualizar informações sobre um ambiente

O comando a seguir recupera informações sobre um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeEnvironments](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para visualizar eventos de um ambiente

O comando a seguir recupera eventos de um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-events --environment-name my-env
```

Saída (abreviada):

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-instances-health

O código de exemplo a seguir mostra como usar `describe-instances-health`.

AWS CLI

Para ver a saúde do meio ambiente

O comando a seguir recupera informações de saúde de instâncias em um ambiente chamado `my-env`:

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --  
attribute-names ALL
```

Saída:

```
{  
  "InstanceHealthList": [  
    {  
      "InstanceId": "i-08691cc7",  
      "ApplicationMetrics": {  
        "Duration": 10,  
        "Latency": {  
          "P99": 0.006,  
          "P75": 0.002,  
          "P90": 0.004,  
          "P95": 0.005,  
          "P85": 0.003,  
          "P10": 0.0,  
          "P999": 0.006,  
          "P50": 0.001  
        },  
        "RequestCount": 48,  
        "StatusCodes": {  
          "Status3xx": 0,  
          "Status2xx": 47,  
          "Status5xx": 0,  
          "Status4xx": 1  
        }  
      },  
    },  
  ],  
}
```

```
    "System": {
      "LoadAverage": [
        0.0,
        0.02,
        0.05
      ],
      "CPUUtilization": {
        "SoftIRQ": 0.1,
        "IOWait": 0.2,
        "System": 0.3,
        "Idle": 97.8,
        "User": 1.5,
        "IRQ": 0.0,
        "Nice": 0.1
      }
    },
    "Color": "Green",
    "HealthStatus": "Ok",
    "LaunchedAt": "2015-08-13T19:17:09Z",
    "Causes": []
  }
],
"RefreshedAt": "2015-08-20T21:09:08Z"
}
```

As informações de saúde só estão disponíveis para ambientes com relatórios de saúde aprimorados habilitados. Para obter mais informações, consulte [Enhanced Health Reporting and Monitoring](#) no Guia do Desenvolvedor do AWS Elastic Beanstalk.

- Para API obter detalhes, consulte [DescribeInstancesHealth](#) na Referência de AWS CLI Comandos.

list-available-solution-stacks

O código de exemplo a seguir mostra como usar `list-available-solution-stacks`.

AWS CLI

Para ver as pilhas de soluções

O comando a seguir lista as pilhas de soluções para todas as configurações de plataforma disponíveis atualmente e todas as que você usou no passado:

aws elasticbeanstalk list-available-solution-stacks

Saída (abreviada):

```
{
  "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"
      },
      ...
    ]
  }
```

- Para API obter detalhes, consulte [ListAvailableSolutionStacks](#) na Referência de AWS CLI Comandos.

rebuild-environment

O código de exemplo a seguir mostra como usar `rebuild-environment`.

AWS CLI

Para reconstruir um ambiente

O comando a seguir encerra e recria os recursos em um ambiente chamado: `my-env`

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- Para API obter detalhes, consulte [RebuildEnvironment](#) na Referência de AWS CLI Comandos.

request-environment-info

O código de exemplo a seguir mostra como usar `request-environment-info`.

AWS CLI

Para solicitar registros com falha

O comando a seguir solicita registros de um ambiente chamado `my-env`:

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-type tail
```

Depois de solicitar os registros, recupere a localização deles com `retrieve-environment-info`

- Para API obter detalhes, consulte [RequestEnvironmentInfo](#) na Referência de AWS CLI Comandos.

restart-app-server

O código de exemplo a seguir mostra como usar `restart-app-server`.

AWS CLI

Para reiniciar os servidores de aplicativos

O comando a seguir reinicia os servidores de aplicativos em todas as instâncias em um ambiente chamado `my-env`:

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- Para API obter detalhes, consulte [RestartAppServer](#) na Referência de AWS CLI Comandos.

retrieve-environment-info

O código de exemplo a seguir mostra como usar `retrieve-environment-info`.

AWS CLI

Para recuperar registros com falha

O comando a seguir recupera um link para registros de um ambiente chamado `my-env`:

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-type tail
```

Saída:

```
{
  "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=%2BEa10V6A2HI0x4Rcfb7LT16bBM%3D",
      "InfoType": "tail",
      "Ec2InstanceId": "i-09c1c867"
```

```
}  
  ]  
}
```

Visualize o link em um navegador. Antes da recuperação, os registros devem ser solicitados com `request-environment-info`.

- Para API obter detalhes, consulte [RetrieveEnvironmentInfo](#) na Referência de AWS CLI Comandos.

swap-environment-cnames

O código de exemplo a seguir mostra como usar `swap-environment-cnames`.

AWS CLI

Para trocar o ambiente CNAMEs

O comando a seguir troca os subdomínios atribuídos de dois ambientes:

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue  
--destination-environment-name my-env-green
```

- Para API obter detalhes, consulte [SwapEnvironmentCnames](#) na Referência de AWS CLI Comandos.

terminate-environment

O código de exemplo a seguir mostra como usar `terminate-environment`.

AWS CLI

Para encerrar um ambiente

O comando a seguir encerra um ambiente do Elastic Beanstalk chamado: `my-env`

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

Saída:

```
{
```

```

    "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"
  }

```

- Para API obter detalhes, consulte [TerminateEnvironment](#) na Referência de AWS CLI Comandos.

update-application-version

O código de exemplo a seguir mostra como usar `update-application-version`.

AWS CLI

Para alterar a descrição de uma versão do aplicativo

O comando a seguir atualiza a descrição de uma versão do aplicativo chamada `22a0-stage-150819_185942`:

```

aws elasticbeanstalk update-application-version --version-label 22a0-
stage-150819_185942 --application-name my-app --description "new description"

```

Saída:

```

{
  "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"
    }
  }
}

```

- Para API obter detalhes, consulte [UpdateApplicationVersion](#) na Referência de AWS CLI Comandos.

update-application

O código de exemplo a seguir mostra como usar `update-application`.

AWS CLI

Para alterar a descrição de um aplicativo

O comando a seguir atualiza a descrição de um aplicativo chamado `my-app`:

```
aws elasticbeanstalk update-application --application-name my-app --description "my Elastic Beanstalk application"
```

Saída:

```

{
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [UpdateApplication](#) na Referência de AWS CLI Comandos.

update-configuration-template

O código de exemplo a seguir mostra como usar `update-configuration-template`.

AWS CLI

Para atualizar um modelo de configuração

O comando a seguir remove a configuração de métricas de saúde CloudWatch personalizadas configuradas `ConfigDocument` de um modelo de configuração salvo chamado `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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre namespaces e opções compatíveis, consulte Valores de opção no Guia do desenvolvedor do Elastic AWS Beanstalk.

- Para API obter detalhes, consulte [UpdateConfigurationTemplate](#) na Referência de AWS CLI Comandos.

update-environment

O código de exemplo a seguir mostra como usar `update-environment`.

AWS CLI

Para atualizar um ambiente para uma nova versão

O comando a seguir atualiza um ambiente chamado “my-env” para a versão “v2” do aplicativo ao qual ele pertence:

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

Esse comando exige que o ambiente “my-env” já exista e pertença a um aplicativo que tenha uma versão válida do aplicativo com o rótulo “v2”.

Saída:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v2",
  "Status": "Updating",
  "EnvironmentId": "e-szqipays4h",
  "EndpointURL": "awseb-e-i-AWSEBLoa-1RDLX6TC9VUA0-0123456789.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"
}
```

Para definir uma variável de ambiente

O comando a seguir define o valor da variável "PARAM1" no ambiente “my-env” como “”:
ParamValue

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para
```

O `option-settings` parâmetro usa um namespace além do nome e do valor da variável. O Elastic Beanstalk oferece suporte a vários namespaces para opções, além de variáveis de ambiente.

Para definir as configurações das opções a partir de um arquivo

O comando a seguir configura várias opções no `aws:elb:loadbalancer` namespace de um arquivo:

```
aws elasticbeanstalk update-environment --environment-name my-env --option-  
settings file://options.json
```

`options.json` é um JSON objeto que define várias configurações:

```
[  
  {  
    "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"  
  }  
]
```

Saída:

```
{  
  "ApplicationName": "my-app",  
  "EnvironmentName": "my-env",
```



```
"VersionLabel": "7f58-stage-150812_025409",
"Status": "Updating",
"EnvironmentId": "e-wtp2rqpqsej",
"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"
}
```

Para obter mais informações sobre namespaces e opções compatíveis, consulte Valores de opção no Guia do desenvolvedor do Elastic AWS Beanstalk.

- Para API obter detalhes, consulte [UpdateEnvironment](#) na Referência de AWS CLI Comandos.

validate-configuration-settings

O código de exemplo a seguir mostra como usar `validate-configuration-settings`.

AWS CLI

Para validar as configurações

O comando a seguir valida um documento de configuração de métricas CloudWatch personalizadas:

```
aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings file://options.json
```

`options.json` é um JSON documento que inclui uma ou mais definições de configuração para validar:

```
[
```

```

{
  "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 as opções que você especificar forem válidas para o ambiente especificado, o Elastic Beanstalk retornará uma matriz de mensagens vazia:

```

{
  "Messages": []
}

```

Se a validação falhar, a resposta incluirá informações sobre o erro:

```

{
  "Messages": [
    {
      "OptionName": "ConfigDocumet",
      "Message": "Invalid option specification (Namespace:
'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
Unknown configuration setting.",
      "Namespace": "aws:elasticbeanstalk:healthreporting:system",
      "Severity": "error"
    }
  ]
}

```

```
    }  
  ]  
}
```

Para obter mais informações sobre namespaces e opções compatíveis, consulte Valores de opção no Guia do desenvolvedor do Elastic AWS Beanstalk.

- Para API obter detalhes, consulte [ValidateConfigurationSettings](#) na Referência de AWS CLI Comandos.

Elastic Load Balancing — Exemplos da versão 1 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Elastic Load Balancing - Versão 1.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags

O código de exemplo a seguir mostra como usar `add-tags`.

AWS CLI

Para adicionar uma tag a um balanceador de carga

Este exemplo adiciona tags ao balanceador de carga especificado.

Comando:

```
aws elb add-tags --load-balancer-name my-load-balancer --  
tags "Key=project,Value=Lima" "Key=department,Value=digital-media"
```

- Para API obter detalhes, consulte [AddTags](#) na Referência de AWS CLI Comandos.

apply-security-groups-to-load-balancer

O código de exemplo a seguir mostra como usar `apply-security-groups-to-load-balancer`.

AWS CLI

Para associar um grupo de segurança a um balanceador de carga em um VPC

Este exemplo associa um grupo de segurança ao balanceador de carga especificado em um VPC

Comando:

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer
--security-groups sg-fc448899
```

Saída:

```
{
  "SecurityGroups": [
    "sg-fc448899"
  ]
}
```

- Para API obter detalhes, consulte [ApplySecurityGroupsToLoadBalancer](#) na Referência de AWS CLI Comandos.

attach-load-balancer-to-subnets

O código de exemplo a seguir mostra como usar `attach-load-balancer-to-subnets`.

AWS CLI

Para anexar sub-redes a um balanceador de carga

Este exemplo adiciona a sub-rede especificada ao conjunto de sub-redes configuradas para o balanceador de carga especificado.

Comando:

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --  
subnets subnet-0ecac448
```

Saída:

```
{  
  "Subnets": [  
    "subnet-15aaab61",  
    "subnet-0ecac448"  
  ]  
}
```

- Para API obter detalhes, consulte [AttachLoadBalancerToSubnets](#) na Referência de AWS CLI Comandos.

configure-health-check

O código de exemplo a seguir mostra como usar `configure-health-check`.

AWS CLI

Para especificar as configurações de verificação de integridade de suas instâncias de back-end EC2

Este exemplo especifica as configurações de verificação de integridade usadas para avaliar a integridade de suas instâncias de back-end 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
```

Saída:

```
{  
  "HealthCheck": {  
    "HealthyThreshold": 2,  
    "Interval": 30,  
    "Target": "HTTP:80/png",  
  }  
}
```

```
    "Timeout": 3,  
    "UnhealthyThreshold": 2  
  }  
}
```

- Para API obter detalhes, consulte [ConfigureHealthCheck](#) na Referência de AWS CLI Comandos.

create-app-cookie-stickness-policy

O código de exemplo a seguir mostra como usar `create-app-cookie-stickness-policy`.

AWS CLI

Para gerar uma política de aderência para seu balanceador de HTTPS carga

Este exemplo gera uma política de fidelidade que segue a vida útil da sessão fixa do cookie gerado pelo aplicativo.

Comando:

```
aws elb create-app-cookie-stickness-policy --load-balancer-name my-load-balancer --  
policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- Para API obter detalhes, consulte [CreateAppCookieStickinessPolicy](#) na Referência de AWS CLI Comandos.

create-lb-cookie-stickness-policy

O código de exemplo a seguir mostra como usar `create-lb-cookie-stickness-policy`.

AWS CLI

Para gerar uma política de aderência baseada em duração para seu balanceador de carga HTTPS

Este exemplo gera uma política de permanência com períodos de vida de sessão fixos controlados pelo período de expiração especificado.

Comando:

```
aws elb create-lb-cookie-stickiness-policy --load-balancer-name my-load-balancer --  
policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- Para API obter detalhes, consulte [CreateLbCookieStickinessPolicy](#) na Referência de AWS CLI Comandos.

create-load-balancer-listeners

O código de exemplo a seguir mostra como usar `create-load-balancer-listeners`.

AWS CLI

Para criar HTTP ouvintes para um balanceador de carga

Este exemplo cria um ouvinte para seu balanceador de carga na porta 80 usando o HTTP protocolo.

Comando:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

Para criar HTTPS ouvintes para um balanceador de carga

Este exemplo cria um ouvinte para seu balanceador de carga na porta 443 usando o protocolo. HTTPS

Comando:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- Para API obter detalhes, consulte [CreateLoadBalancerListeners](#) na Referência de AWS CLI Comandos.

create-load-balancer-policy

O código de exemplo a seguir mostra como usar `create-load-balancer-policy`.

AWS CLI

Para criar uma política que habilite o Proxy Protocol em um balanceador de carga

Este exemplo cria uma política que ativa o Proxy Protocol no balanceador de carga especificado.

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
```

Para criar uma política SSL de negociação usando a política de segurança recomendada

Este exemplo cria uma política de SSL negociação para o balanceador de HTTPS carga especificado usando a política de segurança recomendada.

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
```

Para criar uma política SSL de negociação usando uma política de segurança personalizada

Este exemplo cria uma política de SSL negociação para seu balanceador de HTTPS carga usando uma política de segurança personalizada, ativando os protocolos e as cifras.

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
```

Para criar uma política de chave pública

Este exemplo cria uma política de chave pública.

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=MIIBIjANBgkqhkiG9w0BAQEFAAAOCAQ8AMIIBCgKCAQ
+dS74kj//c6x7R0tusUaeQCTgIUkayttRDWchuqo1pHC1u
+n5xxXnBBE2ejbb2WRsKIQ5rXEeixsjFpFsojpSQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe
+o4haAWvDBALJU0pkSI1FecPHcs2hwxf14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCE0jnYRg7RPvybaGfQ
+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPubcUkkXDj45vC2s3Z8E
+Lk7a3uZhvsQHLZnrfuWjBWGWvZ/MhZYgEXAMPLE
```

Para criar uma política de autenticação de servidor de back-end

Este exemplo cria uma política de autenticação de servidor de back-end que permite a autenticação em sua instância de back-end usando uma política de chave pública.

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
```

- Para API obter detalhes, consulte [CreateLoadBalancerPolicy](#) na Referência de AWS CLI Comandos.

create-load-balancer

O código de exemplo a seguir mostra como usar create-load-balancer.

AWS CLI

Para criar um HTTP balanceador de carga

Este exemplo cria um balanceador de carga com um HTTP ouvinte em um VPC

Comando:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
--subnets subnet-15aab61 --security-groups sg-a61988c3
```

Saída:

```
{
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

Este exemplo cria um balanceador de carga com um HTTP ouvinte em EC2 -Classic.

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
```

Saída:

```
{
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

Para criar um HTTPS balanceador de carga

Este exemplo cria um balanceador de carga com um HTTPS ouvinte em um 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-15aaab61 --security-groups sg-a61988c3
```

Saída:

```
{
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

Este exemplo cria um balanceador de carga com um HTTPS ouvinte em EC2 -Classic.

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
```

Saída:

```
{
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

Para criar um balanceador de carga interno

Este exemplo cria um balanceador de carga interno com um HTTP ouvinte em um 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
```

Saída:

```
{
  "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

- Para API obter detalhes, consulte [CreateLoadBalancer](#) na Referência de AWS CLI Comandos.

delete-load-balancer-listeners

O código de exemplo a seguir mostra como usar `delete-load-balancer-listeners`.

AWS CLI

Para excluir um ouvinte do seu balanceador de carga

Este exemplo exclui o ouvinte da porta especificada do balanceador de carga especificado.

Comando:

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-
balancer-ports 80
```

- Para API obter detalhes, consulte [DeleteLoadBalancerListeners](#) na Referência de AWS CLI Comandos.

delete-load-balancer-policy

O código de exemplo a seguir mostra como usar `delete-load-balancer-policy`.

AWS CLI

Para excluir uma política do seu balanceador de carga

Este exemplo exclui a política especificada do balanceador de carga especificado. A política não deve ser ativada em nenhum ouvinte.

Comando:

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- Para API obter detalhes, consulte [DeleteLoadBalancerPolicy](#) na Referência de AWS CLI Comandos.

delete-load-balancer

O código de exemplo a seguir mostra como usar `delete-load-balancer`.

AWS CLI

Como excluir um balanceador de carga

Este exemplo exclui o balanceador de carga especificado.

Comando:

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- Para API obter detalhes, consulte [DeleteLoadBalancer](#) na Referência de AWS CLI Comandos.

deregister-instances-from-load-balancer

O código de exemplo a seguir mostra como usar `deregister-instances-from-load-balancer`.

AWS CLI

Para cancelar o registro de instâncias de um balanceador de carga

Este exemplo cancela o registro da instância especificada do balanceador de carga especificado.

Comando:

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-balancer --instances i-d6f6fae3
```

Saída:

```
{
  "Instances": [
    {
      "InstanceId": "i-207d9717"
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

- Para API obter detalhes, consulte [DeregisterInstancesFromLoadBalancer](#) na Referência de AWS CLI Comandos.

describe-account-limits

O código de exemplo a seguir mostra como usar `describe-account-limits`.

AWS CLI

Para descrever seus limites do Classic Load Balancer

O `describe-account-limits` exemplo a seguir exibe detalhes sobre os limites do Classic Load Balancer para sua AWS conta.

```
aws elb describe-account-limits
```

Saída:

```
{
  "Limits": [
    {
      "Name": "classic-load-balancers",
      "Max": "20"
    },
    {
      "Name": "classic-listeners",
      "Max": "100"
    },
    {
      "Name": "classic-registered-instances",
      "Max": "1000"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeAccountLimits](#) na Referência de AWS CLI Comandos.

describe-instance-health

O código de exemplo a seguir mostra como usar `describe-instance-health`.

AWS CLI

Para descrever a integridade das instâncias de um balanceador de carga

Este exemplo descreve a integridade das instâncias do balanceador de carga especificado.

Comando:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

Saída:

```
{
  "InstanceStates": [
    {
      "InstanceId": "i-207d9717",
      "ReasonCode": "N/A",
      "State": "InService",
```

```
    "Description": "N/A"
  },
  {
    "InstanceId": "i-afefb49b",
    "ReasonCode": "N/A",
    "State": "InService",
    "Description": "N/A"
  }
]
}
```

Para descrever a integridade de uma instância para um balanceador de carga

Este exemplo descreve a integridade da instância especificada para o balanceador de carga especificado.

Comando:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --
instances i-7299c809
```

Veja a seguir um exemplo de resposta para uma instância que está sendo registrada.

Saída:

```
{
  "InstanceStates": [
    {
      "InstanceId": "i-7299c809",
      "ReasonCode": "ELB",
      "State": "OutOfService",
      "Description": "Instance registration is still in progress."
    }
  ]
}
```

Veja a seguir um exemplo de resposta para uma instância não íntegra.

Saída:

```
{
  "InstanceStates": [
    {
```

```
        "InstanceId": "i-7299c809",
        "ReasonCode": "Instance",
        "State": "OutOfService",
        "Description": "Instance has failed at least the UnhealthyThreshold number
of health checks consecutively."
    }
]
}
```

- Para API obter detalhes, consulte [DescribeInstanceHealth](#) na Referência de AWS CLI Comandos.

describe-load-balancer-attributes

O código de exemplo a seguir mostra como usar `describe-load-balancer-attributes`.

AWS CLI

Para descrever os atributos de um balanceador de carga

Este exemplo descreve os atributos do balanceador de carga especificado.

Comando:

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

Saída:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": false,
      "Timeout": 300
    },
    "CrossZoneLoadBalancing": {
      "Enabled": true
    },
    "ConnectionSettings": {
      "IdleTimeout": 30
    },
    "AccessLog": {
      "Enabled": false
    }
  }
}
```



```
}  
}  
}
```

- Para API obter detalhes, consulte [DescribeLoadBalancerAttributes](#) na Referência de AWS CLI Comandos.

describe-load-balancer-policies

O código de exemplo a seguir mostra como usar `describe-load-balancer-policies`.

AWS CLI

Para descrever todas as políticas associadas a um balanceador de carga

Este exemplo descreve todas as políticas associadas ao balanceador de carga especificado.

Comando:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

Saída:

```
{  
  "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"
  },
  .
  .
  .
]
}

```

Para descrever uma política específica associada a um balanceador de carga

Este exemplo descreve a política especificada associada ao balanceador de carga especificado.

Comando:

```

aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --
policy-name my-authentication-policy

```

Saída:

```

{
  "PolicyDescriptions": [
    {
      "PolicyAttributeDescriptions": [
        {
          "AttributeName": "PublicKeyPolicyName",
          "AttributeValue": "my-PublicKey-policy"
        }
      ],
      "PolicyName": "my-authentication-policy",
      "PolicyTypeName": "BackendServerAuthenticationPolicyType"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeLoadBalancerPolicies](#) na Referência de AWS CLI Comandos.

describe-load-balancer-policy-types

O código de exemplo a seguir mostra como usar `describe-load-balancer-policy-types`.

AWS CLI

Para descrever os tipos de política do balanceador de carga definidos pelo Elastic Load Balancing

Este exemplo descreve os tipos de política do balanceador de carga que você pode usar para criar configurações de política para seu balanceador de carga.

Comando:

```
aws elb describe-load-balancer-policy-types
```

Saída:

```
{
  "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."
}
]
```

- Para API obter detalhes, consulte [DescribeLoadBalancerPolicyTypes](#) na Referência de AWS CLI Comandos.

describe-load-balancers

O código de exemplo a seguir mostra como usar `describe-load-balancers`.

AWS CLI

Para descrever seus balanceadores de carga

Este exemplo descreve todos os balanceadores de carga.

Comando:

```
aws elb describe-load-balancers
```

Para descrever um de seus balanceadores de carga

Este exemplo descreve o balanceador de carga especificado.

Comando:

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```

O exemplo de resposta a seguir é para um balanceador de HTTPS carga em um VPC.

Saída:

```
{
  "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"
}
```

```
    }  
  }  
]  
}
```

- Para API obter detalhes, consulte [DescribeLoadBalancers](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever as tags atribuídas a um balanceador de carga

Este exemplo descreve as tags atribuídas ao balanceador de carga especificado.

Comando:

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

Saída:

```
{  
  "TagDescriptions": [  
    {  
      "Tags": [  
        {  
          "Value": "lima",  
          "Key": "project"  
        },  
        {  
          "Value": "digital-media",  
          "Key": "department"  
        }  
      ],  
      "LoadBalancerName": "my-load-balancer"  
    }  
  ]  
}
```


- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

detach-load-balancer-from-subnets

O código de exemplo a seguir mostra como usar `detach-load-balancer-from-subnets`.

AWS CLI

Para separar os balanceadores de carga das sub-redes

Este exemplo separa o balanceador de carga especificado da sub-rede especificada.

Comando:

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --  
subnets subnet-0ecac448
```

Saída:

```
{  
  "Subnets": [  
    "subnet-15aaab61"  
  ]  
}
```

- Para API obter detalhes, consulte [DetachLoadBalancerFromSubnets](#) na Referência de AWS CLI Comandos.

disable-availability-zones-for-load-balancer

O código de exemplo a seguir mostra como usar `disable-availability-zones-for-load-balancer`.

AWS CLI

Para desativar as zonas de disponibilidade para um balanceador de carga

Este exemplo remove a zona de disponibilidade especificada do conjunto de zonas de disponibilidade do balanceador de carga especificado.

Comando:

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-Load-balancer --availability-zones us-west-2a
```

Saída:

```
{
  "AvailabilityZones": [
    "us-west-2b"
  ]
}
```

- Para API obter detalhes, consulte [DisableAvailabilityZonesForLoadBalancer](#) na Referência de AWS CLI Comandos.

enable-availability-zones-for-load-balancer

O código de exemplo a seguir mostra como usar `enable-availability-zones-for-load-balancer`.

AWS CLI

Para habilitar zonas de disponibilidade para um balanceador de carga

Este exemplo adiciona a zona de disponibilidade especificada ao balanceador de carga especificado.

Comando:

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-Load-balancer --availability-zones us-west-2b
```

Saída:

```
{
  "AvailabilityZones": [
    "us-west-2a",
    "us-west-2b"
  ]
}
```

- Para API obter detalhes, consulte [EnableAvailabilityZonesForLoadBalancer](#) na Referência de AWS CLI Comandos.

modify-load-balancer-attributes

O código de exemplo a seguir mostra como usar `modify-load-balancer-attributes`.

AWS CLI

Para modificar os atributos de um balanceador de carga

Este exemplo modifica o `CrossZoneLoadBalancing` atributo do balanceador de carga especificado.

Comando:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

Saída:

```
{
  "LoadBalancerAttributes": {
    "CrossZoneLoadBalancing": {
      "Enabled": true
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

Este exemplo modifica o `ConnectionDraining` atributo do balanceador de carga especificado.

Comando:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout\":300}}"
```

Saída:

```
{
```

```
"LoadBalancerAttributes": {
  "ConnectionDraining": {
    "Enabled": true,
    "Timeout": 300
  }
},
"LoadBalancerName": "my-load-balancer"
}
```

- Para API obter detalhes, consulte [ModifyLoadBalancerAttributes](#) na Referência de AWS CLI Comandos.

register-instances-with-load-balancer

O código de exemplo a seguir mostra como usar `register-instances-with-load-balancer`.

AWS CLI

Para registrar instâncias com um balanceador de carga

Este exemplo registra a instância especificada com o balanceador de carga especificado.

Comando:

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
--instances i-d6f6fae3
```

Saída:

```
{
  "Instances": [
    {
      "InstanceId": "i-d6f6fae3"
    },
    {
      "InstanceId": "i-207d9717"
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [RegisterInstancesWithLoadBalancer](#) na Referência de AWS CLI Comandos.

remove-tags

O código de exemplo a seguir mostra como usar `remove-tags`.

AWS CLI

Para remover tags de um balanceador de carga

Este exemplo remove uma tag do balanceador de carga especificado.

Comando:

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- Para API obter detalhes, consulte [RemoveTags](#) na Referência de AWS CLI Comandos.

set-load-balancer-listener-ssl-certificate

O código de exemplo a seguir mostra como usar `set-load-balancer-listener-ssl-certificate`.

AWS CLI

Para atualizar o SSL certificado de um HTTPS balanceador de carga

Este exemplo substitui o SSL certificado existente para o HTTPS balanceador de carga especificado.

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
```

- Para API obter detalhes, consulte [SetLoadBalancerListenerSslCertificate](#) na Referência de AWS CLI Comandos.

set-load-balancer-policies-for-backend-server

O código de exemplo a seguir mostra como usar `set-load-balancer-policies-for-backend-server`.

AWS CLI

Para substituir as políticas associadas a uma porta para uma instância de back-end

Este exemplo substitui as políticas atualmente associadas à porta especificada.

Comando:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

Para remover todas as políticas atualmente associadas a uma porta na sua instância de back-end

Este exemplo remove todas as políticas associadas à porta especificada.

Comando:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names []
```

Para confirmar que as políticas foram removidas, use o `describe-load-balancer-policies` comando.

- Para API obter detalhes, consulte [SetLoadBalancerPoliciesForBackendServer](#) na Referência de AWS CLI Comandos.

set-load-balancer-policies-of-listener

O código de exemplo a seguir mostra como usar `set-load-balancer-policies-of-listener`.

AWS CLI

Para substituir as políticas associadas a um ouvinte

Este exemplo substitui as políticas atualmente associadas ao ouvinte especificado.

Comando:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
--load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

Para remover todas as políticas associadas ao seu ouvinte

Este exemplo remove todas as políticas atualmente associadas ao ouvinte especificado.

Comando:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
--load-balancer-port 443 --policy-names []
```

Para confirmar se as políticas foram removidas do balanceador de carga, use o `describe-load-balancer-policies` comando.

- Para API obter detalhes, consulte [SetLoadBalancerPoliciesOfListener](#) na Referência de AWS CLI Comandos.

Elastic Load Balancing - Exemplos da versão 2 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Elastic Load Balancing - Versão 2.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-listener-certificates

O código de exemplo a seguir mostra como usar `add-listener-certificates`.

AWS CLI

Para adicionar um certificado a um ouvinte seguro

Este exemplo adiciona o certificado especificado ao ouvinte seguro especificado.

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
```

Saída:

```
{
  "Certificates": [
    {
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
      "IsDefault": false
    }
  ]
}
```

- Para API obter detalhes, consulte [AddListenerCertificates](#) na Referência de AWS CLI Comandos.

add-tags

O código de exemplo a seguir mostra como usar add-tags.

AWS CLI

Para adicionar tags a um balanceador de carga

O add-tags exemplo a seguir adiciona as department tags project and ao balanceador de carga especificado.

```
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"
```

- Para API obter detalhes, consulte [AddTags](#) na Referência de AWS CLI Comandos.

create-listener

O código de exemplo a seguir mostra como usar `create-listener`.

AWS CLI

Exemplo 1: Para criar um HTTP ouvinte

O `create-listener` exemplo a seguir cria um HTTP ouvinte para o Application Load Balancer especificado que encaminha solicitações para o grupo-alvo especificado.

```
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
```

Para obter mais informações, consulte [Tutorial: Crie um Application Load Balancer usando o Guia do AWS CLI](#) usuário para Application Load Balancers.

Exemplo 2: Para criar um HTTPS ouvinte

O `create-listener` exemplo a seguir cria um HTTPS ouvinte para o Application Load Balancer especificado que encaminha solicitações para o grupo-alvo especificado. Você deve especificar um SSL certificado para um HTTPS ouvinte. Você pode criar e gerenciar certificados usando o AWS Certificate Manager (ACM). Como alternativa, você pode criar um certificado usando SSL/TLS tools, obter o certificado assinado por uma autoridade de certificação (CA) e fazer o upload do certificado para 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 arn:aws:iam::123456789012:certificate/12345678-1234-1234-1234-123456789012
```

```

--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

```

Para obter mais informações, consulte [Adicionar um HTTPS ouvinte](#) no Guia do usuário para balanceadores de carga de aplicativos.

Exemplo 3: Para criar um TCP ouvinte

O `create-listener` exemplo a seguir cria um TCP ouvinte para o Network Load Balancer especificado que encaminha solicitações para o grupo-alvo especificado.

```

aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/net/my-network-load-balancer/5d1b75f4f1cee11e \
  --protocol TCP \
  --port 80 \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78

```

Para obter mais informações, consulte [Tutorial: Crie um Network Load Balancer usando o AWS CLI](#) Guia do usuário para Network Load Balancers.

Exemplo 4: Para criar um TLS ouvinte

O `create-listener` exemplo a seguir cria um TLS ouvinte para o Network Load Balancer especificado que encaminha solicitações para o grupo-alvo especificado. Você deve especificar um SSL certificado para um TLS ouvinte.

```

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

```

Para obter mais informações, consulte [TLSouvintes do seu Network Load Balancer](#) no Guia do usuário para Network Load Balancers.

Exemplo 5: Para criar um UDP ouvinte

O `create-listener` exemplo a seguir cria um UDP ouvinte para o Network Load Balancer especificado que encaminha solicitações para o grupo-alvo especificado.

```
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
```

Para obter mais informações, consulte [Tutorial: Crie um Network Load Balancer usando o AWS CLI](#) Guia do usuário para Network Load Balancers.

Exemplo 6: como criar um receptor para o gateway e encaminhamento especificados

O exemplo de `create-listener` a seguir cria um receptor para o Gateway Load Balancer especificado. Esse receptor encaminha as solicitações para o grupo de destino especificado.

```
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
```

Saída:

```
{
  "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"
          }
        ]
      }
    }
  ]
}

```

Para obter mais informações, consulte [Introdução aos balanceadores de carga de gateway usando o AWS CLI](#) Guia do usuário para balanceadores de carga de gateway.

- Para API obter detalhes, consulte [CreateListener](#) na Referência de AWS CLI Comandos.

create-load-balancer

O código de exemplo a seguir mostra como usar `create-load-balancer`.

AWS CLI

Exemplo 1: como criar um balanceador de carga voltado para a internet

O exemplo de `create-load-balancer` a seguir cria um Application Load Balancer voltado para a internet e habilita as zonas de disponibilidade para as sub-redes especificadas.

```

aws elbv2 create-load-balancer \
  --name my-load-balancer \
  --subnets subnet-b7d581c0 subnet-8360a9e7

```

Saída:

```

{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Tutorial: Crie um Application Load Balancer usando o Guia do AWS CLI](#) usuário para Application Load Balancers.

Exemplo 2: como criar um balanceador de carga interno

O exemplo de `create-load-balancer` a seguir cria um Application Load Balancer interno e habilita as zonas de disponibilidade para as sub-redes especificadas.

```

aws elbv2 create-load-balancer \
  --name my-internal-load-balancer \
  --scheme internal \
  --subnets subnet-b7d581c0 subnet-8360a9e7

```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Tutorial: Crie um Application Load Balancer usando o Guia do AWS CLI](#) usuário para Application Load Balancers.

Exemplo 3: como criar um Network Load Balancer

O exemplo de `create-load-balancer` a seguir cria um Network Load Balancer voltado para a internet e habilita a zona de disponibilidade para as sub-redes especificadas. Ele usa o

mapeamento de sub-redes para associar o endereço IP do Elastic específico à interface de rede usada pelos nós do balanceador de carga da zona de disponibilidade.

```
aws elbv2 create-load-balancer \  
  --name my-network-load-balancer \  
  --type network \  
  --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Tutorial: Crie um Network Load Balancer usando o AWS CLI](#) Guia do usuário para Network Load Balancers.

Exemplo 4: como criar um Gateway Load Balancer

O exemplo de `create-load-balancer` a seguir cria um Gateway Load Balancer e habilita as zonas de disponibilidade para as sub-redes especificadas.

```
aws elbv2 create-load-balancer \  
  --name my-gateway-load-balancer \  
  --type gateway \  
  --subnets subnet-dc83f691 subnet-a62583f9
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```


Para obter mais informações, consulte [Introdução aos balanceadores de carga de gateway usando o AWS CLI](#) Guia do usuário para balanceadores de carga de gateway.

- Para API obter detalhes, consulte [CreateLoadBalancer](#) na Referência de AWS CLI Comandos.

create-rule

O código de exemplo a seguir mostra como usar `create-rule`.

AWS CLI

Exemplo 1: Para criar uma regra usando uma condição de caminho e uma ação de avanço

O `create-rule` exemplo a seguir cria uma regra que encaminha solicitações para o grupo-alvo especificado se ele URL contiver o padrão especificado.

```
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
```

Conteúdo de `conditions-pattern.json`:

```
[  
  {  
    "Field": "path-pattern",  
    "PathPatternConfig": {  
      "Values": ["/images/*"]  
    }  
  }  
]
```

Exemplo 2: Para criar uma regra usando uma condição de host e uma resposta fixa

O `create-rule` exemplo a seguir cria uma regra que fornece uma resposta fixa se o nome do host no cabeçalho do host corresponder ao nome do host especificado.

```
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
```

```
--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
```

Conteúdo de `conditions-host.json`

```
[  
  {  
    "Field": "host-header",  
    "HostHeaderConfig": {  
      "Values": ["*.example.com"]  
    }  
  }  
]
```

Conteúdo de `actions-fixed-response.json`

```
[  
  {  
    "Type": "fixed-response",  
    "FixedResponseConfig": {  
      "MessageBody": "Hello world",  
      "StatusCode": "200",  
      "ContentType": "text/plain"  
    }  
  }  
]
```

Exemplo 3: Para criar uma regra usando uma condição de endereço IP de origem, uma ação de autenticação e uma ação de encaminhamento

O `create-rule` exemplo a seguir cria uma regra que autentica o usuário se o endereço IP de origem corresponder ao endereço IP especificado e encaminha a solicitação para o grupo-alvo especificado se a autenticação for bem-sucedida.

```
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
```

Conteúdo de conditions-source-ip.json

```
[  
  {  
    "Field": "source-ip",  
    "SourceIpConfig": {  
      "Values": ["192.0.2.0/24", "198.51.100.10/32"]  
    }  
  }  
]
```

Conteúdo de 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  
  }  
]
```

```
]
```

- Para API obter detalhes, consulte [CreateRule](#) na Referência de AWS CLI Comandos.

create-target-group

O código de exemplo a seguir mostra como usar `create-target-group`.

AWS CLI

Exemplo 1: Para criar um grupo-alvo para um Application Load Balancer

O exemplo de `create-target-group` a seguir cria um grupo de destino para um Application Load Balancer no qual você registra destinos por ID de instância (o tipo do destino é `instance`). Esse grupo-alvo usa o HTTP protocolo, a porta 80 e as configurações padrão de verificação de integridade de um HTTP grupo-alvo.

```
aws elbv2 create-target-group \  
  --name my-targets \  
  --protocol HTTP \  
  --port 80 \  
  --target-type instance \  
  --vpc-id vpc-3ac0fb5f
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Criar um grupo de destino](#) no Guia do usuário dos Application Load Balancers.

Exemplo 2: Para criar um grupo-alvo para rotear o tráfego de um Application Load Balancer para uma função Lambda

O exemplo de `create-target-group` a seguir cria um grupo de destino para um Application Load Balancer, no qual o destino é uma função do Lambda (o tipo do destino é `lambda`). Por padrão, as verificações de integridade estão desabilitadas para esse grupo de destino.

```

aws elbv2 create-target-group \
  --name my-lambda-target \
  --target-type lambda

```

Saída:

```

{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Lambda functions as targets](#) no Guia do usuário de Application Load Balancers.

Exemplo 3: Para criar um grupo-alvo para um Network Load Balancer

O exemplo de `create-target-group` a seguir cria um grupo de destino para um Network Load Balancer no qual você registra destinos por endereço IP (o tipo do destino é `ip`). Esse grupo-alvo usa o TCP protocolo, a porta 80 e as configurações padrão de verificação de integridade de um TCP grupo-alvo.

```

aws elbv2 create-target-group \
  --name my-ip-targets \
  --protocol TCP \
  --port 80 \
  --target-type ip \
  --vpc-id vpc-3ac0fb5f

```

Saída:

```

{
  "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"
    }
]
}
```

Para obter mais informações, consulte [Criar um grupo-alvo](#) no Guia do usuário para balanceadores de carga de rede.

Exemplo 4: Para criar um grupo-alvo para rotear o tráfego de um Network Load Balancer para um Application Load Balancer

O `create-target-group` exemplo a seguir cria um grupo-alvo para um Network Load Balancer em que você registra um Application Load Balancer como destino (o tipo de destino é `alb`).

```
aws elbv2 create-target-group --name my-alb-target --protocol TCP --port 80 --target-type alb --vpc-id vpc-3ac0fb5f
```

Saída:

```
{
  "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"
      }
    ]
  }

```

Para obter mais informações, consulte [Criar um grupo-alvo com um Application Load Balancer como destino no](#) Guia do usuário para Network Load Balancers.

Exemplo 5: Para criar um grupo-alvo para um Gateway Load Balancer

O `create-target-group` exemplo a seguir cria um grupo-alvo para um Gateway Load Balancer em que o destino é uma instância e o protocolo do grupo-alvo é. GENEVE

```

aws elbv2 create-target-group \
  --name my-glb-targetgroup \
  --protocol GENEVE \
  --port 6081 \
  --target-type instance \
  --vpc-id vpc-838475fe

```

Saída:

```

{
  "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"
    }
  ]
}

```


Para obter mais informações, consulte Criar um grupo-alvo < <https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html> >`__ no Guia do usuário do Gateway Load Balancer.

- Para API obter detalhes, consulte [CreateTargetGroup](#) na Referência de AWS CLI Comandos.

delete-listener

O código de exemplo a seguir mostra como usar `delete-listener`.

AWS CLI

Para excluir um ouvinte

O `delete-listener` exemplo a seguir exclui o ouvinte especificado.

```
aws elbv2 delete-listener \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- Para API obter detalhes, consulte [DeleteListener](#) na Referência de AWS CLI Comandos.

delete-load-balancer

O código de exemplo a seguir mostra como usar `delete-load-balancer`.

AWS CLI

Como excluir um balanceador de carga

O exemplo `delete-load-balancer` a seguir exclui o balanceador de carga especificado.

```
aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- Para API obter detalhes, consulte [DeleteLoadBalancer](#) na Referência de AWS CLI Comandos.

delete-rule

O código de exemplo a seguir mostra como usar `delete-rule`.

AWS CLI

Para excluir uma regra

O `delete-rule` exemplo a seguir exclui a regra especificada.

```
aws elbv2 delete-rule \  
  --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- Para API obter detalhes, consulte [DeleteRule](#) na Referência de AWS CLI Comandos.

`delete-target-group`

O código de exemplo a seguir mostra como usar `delete-target-group`.

AWS CLI

Como excluir um grupo de destino

O exemplo de `delete-target-group` a seguir exclui o grupo de destino especificado.

```
aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um balanceador de carga no Application Load Balancer](#) Guide.

- Para API obter detalhes, consulte [DeleteTargetGroup](#) na Referência de AWS CLI Comandos.

`deregister-targets`

O código de exemplo a seguir mostra como usar `deregister-targets`.

AWS CLI

Exemplo 1: Para cancelar o registro de um alvo de um grupo-alvo

O `deregister-targets` exemplo a seguir remove a instância especificada do grupo de destino especificado.

```
aws elbv2 deregister-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \  
  --targets Id=i-1234567890abcdef0
```

Exemplo 2: Para cancelar o registro de um destino registrado usando substituições de porta

O `deregister-targets` exemplo a seguir remove uma instância de um grupo de destino que foi registrado usando substituições de 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
```

- Para API obter detalhes, consulte [DeregisterTargets](#) na Referência de AWS CLI Comandos.

describe-account-limits

O código de exemplo a seguir mostra como usar `describe-account-limits`.

AWS CLI

Para descrever seus limites do Elastic Load Balancing

O `describe-account-limits` exemplo a seguir mostra os limites do Elastic Load Balancing para sua AWS conta na região atual.

```
aws elbv2 describe-account-limits
```

Saída:

```
{  
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Cotas](#) na Referência AWS Geral.

- Para API obter detalhes, consulte [DescribeAccountLimits](#) na Referência de AWS CLI Comandos.

describe-listener-certificates

O código de exemplo a seguir mostra como usar `describe-listener-certificates`.

AWS CLI

Para descrever os certificados para um ouvinte seguro

Este exemplo descreve os certificados para o ouvinte seguro especificado.

Comando:

```
aws elbv2 describe-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeListenerCertificates](#) na Referência de AWS CLI Comandos.

describe-listeners

O código de exemplo a seguir mostra como usar `describe-listeners`.

AWS CLI

Para descrever um ouvinte

Este exemplo descreve o ouvinte especificado.

Comando:

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para descrever os ouvintes de um balanceador de carga

Este exemplo descreve os ouvintes do balanceador de carga especificado.

Comando:

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Saída:

```
{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [DescribeListeners](#) na Referência de AWS CLI Comandos.

describe-load-balancer-attributes

O código de exemplo a seguir mostra como usar `describe-load-balancer-attributes`.

AWS CLI

Para descrever os atributos do balanceador de carga

O `describe-load-balancer-attributes` exemplo a seguir exibe os atributos do balanceador de carga especificado.

```

aws elbv2 describe-load-balancer-attributes \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188

```

O exemplo de saída a seguir mostra os atributos de um 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"
  }
]
}
```

O exemplo de saída a seguir inclui os atributos de um 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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeLoadBalancerAttributes](#) na Referência de AWS CLI Comandos.

describe-load-balancers

O código de exemplo a seguir mostra como usar `describe-load-balancers`.

AWS CLI

Como descrever um balanceador de carga

Este exemplo descreve o balanceador de carga especificado.

Comando:

```
aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Saída:

```
{
  "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"
    }
  ]
}
```

Como descrever todos os balanceadores de carga

Este exemplo descreve todos os balanceadores de carga.

Comando:

```
aws elbv2 describe-load-balancers
```

- Para API obter detalhes, consulte [DescribeLoadBalancers](#) na Referência de AWS CLI Comandos.

describe-rules

O código de exemplo a seguir mostra como usar `describe-rules`.

AWS CLI

Exemplo 1: Para descrever uma regra

O `describe-rules` exemplo a seguir exibe detalhes da regra especificada.

```
aws elbv2 describe-rules \  
  --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/  
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

Exemplo 2: Para descrever as regras para um ouvinte

O `describe-rules` exemplo a seguir exibe detalhes das regras do ouvinte especificado. A saída inclui a regra padrão e quaisquer outras regras que você tenha adicionado.

```
aws elbv2 describe-rules \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- Para API obter detalhes, consulte [DescribeRules](#) na Referência de AWS CLI Comandos.

describe-ssl-policies

O código de exemplo a seguir mostra como usar `describe-ssl-policies`.

AWS CLI

Exemplo 1: Para listar as políticas usadas para SSL negociação por tipo de balanceador de carga

O `describe-ssl-policies` exemplo a seguir exibe os nomes das políticas que você pode usar para SSL negociação com um Application Load Balancer. O exemplo usa o `--query` parâmetro para exibir somente os nomes das políticas.

```
aws elbv2 describe-ssl-policies \  
  --load-balancer-type application \  
  --query SslPolicies[*].Name
```

Saída:

```
[  
  "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"  
]
```

Exemplo 2: Para listar as políticas que oferecem suporte a um protocolo específico

O `describe-ssl-policies` exemplo a seguir exibe os nomes das políticas que oferecem suporte ao protocolo TLS 1.3. O exemplo usa o `--query` parâmetro para exibir somente os nomes das políticas.

```
aws elbv2 describe-ssl-policies \
  --load-balancer-type application \
  --query SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name
```

Saída:

```
[
  "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"
]
```

Exemplo 3: Para exibir as cifras de uma política

O `describe-ssl-policies` exemplo a seguir exibe os nomes das cifras da política especificada. O exemplo usa o `--query` parâmetro para exibir somente os nomes das cifras. A primeira cifra da lista tem prioridade 1 e as cifras restantes estão em ordem de prioridade.

```
aws elbv2 describe-ssl-policies \
  --names ELBSecurityPolicy-TLS13-1-2-2021-06 \
  --query SslPolicies[*].Ciphers[*].Name
```

Saída:

```
[
  "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"  
]
```

Para obter mais informações, consulte [Políticas de segurança](#) no Guia do usuário para balanceadores de carga de aplicativos.

- Para API obter detalhes, consulte [DescribeSslPolicies](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever as tags atribuídas a um balanceador de carga

Este exemplo descreve as tags atribuídas ao balanceador de carga especificado.

Comando:

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```

```

    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

describe-target-group-attributes

O código de exemplo a seguir mostra como usar `describe-target-group-attributes`.

AWS CLI

Para descrever os atributos do grupo-alvo

O `describe-target-group-attributes` exemplo a seguir exibe os atributos do grupo-alvo especificado.

```

aws elbv2 describe-target-group-attributes \
  --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067

```

A saída inclui os atributos se o protocolo for HTTP ou HTTPS e o tipo de destino for `instance` ou `ip`.

```

{
  "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"
    }
  ]
}
```

A saída a seguir inclui os atributos se o protocolo for HTTP ou HTTPS e o tipo de destino `for_lambda`.

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "lambda.multi_value_headers.enabled"
    }
  ]
}
```

A saída a seguir inclui os atributos se o protocolo for TCPTLS,UDP, ou TCP_UDP.

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "proxy_protocol_v2.enabled"
    },
    {
      "Value": "300",
      "Key": "deregistration_delay.timeout_seconds"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeTargetGroupAttributes](#) na Referência de AWS CLI Comandos.

describe-target-groups

O código de exemplo a seguir mostra como usar `describe-target-groups`.

AWS CLI

Exemplo 1: como descrever um grupo de destino

O exemplo de `describe-target-groups` a seguir exibe os detalhes do grupo de destino especificado.

```
aws elbv2 describe-target-groups \  
  --target-group-arns arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Exemplo 2: como descrever todos os grupos de destino de um balanceador de carga

O exemplo de `describe-target-groups` a seguir exibe os detalhes de todos os grupos de destino do balanceador de carga especificado. O exemplo usa o `--query` parâmetro para exibir somente os nomes dos grupos de destino.

```
aws elbv2 describe-target-groups \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --query TargetGroups[*].TargetGroupName
```

Saída:

```
[  
  "my-instance-targets",  
  "my-ip-targets",  
  "my-lambda-target"  
]
```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do Application Load Balancers.

- Para API obter detalhes, consulte [DescribeTargetGroups](#) na Referência de AWS CLI Comandos.

describe-target-health

O código de exemplo a seguir mostra como usar `describe-target-health`.

AWS CLI

Exemplo 1: como descrever a integridade dos destinos de um grupo de destino

O exemplo de `describe-target-health` a seguir exibe os detalhes de integridade dos destinos do grupo de destino especificado. Os destinos estão íntegros.

```
aws elbv2 describe-target-health \  
  --target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Saída:

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-cedddcd4d",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    },
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    }
  ]
}
```

Exemplo 2: como descrever a integridade de um destino

O exemplo de `describe-target-health` a seguir exibe os detalhes de integridade do destino especificado. Este destino é íntegro.

```
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
```

Saída:

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
```

```

        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    }
  ]
}

```

O exemplo de saída a seguir é de um destino cujo grupo de destino não está especificado em uma ação de um receptor. Esse destino não pode receber tráfego do balanceador de carga.

```

{
  "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"
      }
    }
  ]
}

```

O exemplo de saída a seguir é de um destino cujo grupo de destino foi especificado em uma ação de um receptor. O destino ainda está sendo registrado.

```

{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {

```

```
        "State": "initial",
        "Reason": "Elb.RegistrationInProgress",
        "Description": "Target registration is in progress"
    }
}
]
```

O exemplo de saída a seguir é de um destino não íntegro.

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "unhealthy",
        "Reason": "Target.Timeout",
        "Description": "Connection to target timed out"
      }
    }
  ]
}
```

O exemplo de saída a seguir é de um destino que é uma função do Lambda e com verificações de integridade desabilitadas.

```
{
  "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"
      }
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [DescribeTargetHealth](#) na Referência de AWS CLI Comandos.

modify-listener

O código de exemplo a seguir mostra como usar `modify-listener`.

AWS CLI

Exemplo 1: Para alterar a ação padrão para uma ação de encaminhamento

O `modify-listener` exemplo a seguir altera a ação padrão (para uma ação de encaminhamento) para o ouvinte especificado.

```
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
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

Exemplo 2: Para alterar a ação padrão para uma ação de redirecionamento

O `modify-listener` exemplo a seguir altera a ação padrão para uma ação de redirecionamento para o ouvinte especificado.

```
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
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Exemplo 3: Para alterar o certificado do servidor

Este exemplo altera o certificado do servidor para o HTTPS ouvinte especificado.

```
aws elbv2 modify-listener \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/  
my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65 \  
  --default-actions Type=redirect,TargetGroupArn=arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-new-targets/2453ed029918f21f
```



```
--certificates CertificateArn=arn:aws:iam::123456789012:server-certificate/my-new-server-cert
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ModifyListener](#) na Referência de AWS CLI Comandos.

modify-load-balancer-attributes

O código de exemplo a seguir mostra como usar `modify-load-balancer-attributes`.

AWS CLI

Para ativar a proteção contra exclusão

Este exemplo ativa a proteção contra exclusão do balanceador de carga especificado.

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
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para alterar o tempo limite de inatividade

Este exemplo altera o valor do tempo limite de inatividade do balanceador de carga especificado.

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
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para ativar os registros de acesso

Este exemplo habilita registros de acesso para o balanceador de carga especificado. Observe que o bucket do S3 deve existir na mesma região do balanceador de carga e deve ter uma política anexada que conceda acesso ao serviço 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
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ModifyLoadBalancerAttributes](#) na Referência de AWS CLI Comandos.

modify-rule

O código de exemplo a seguir mostra como usar `modify-rule`.

AWS CLI

Para modificar uma regra

O `modify-rule` exemplo a seguir atualiza as ações e condições da regra especificada.

```
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
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [ModifyRule](#) na Referência de AWS CLI Comandos.

modify-target-group-attributes

O código de exemplo a seguir mostra como usar `modify-target-group-attributes`.

AWS CLI

Para modificar o tempo limite de atraso do cancelamento de registro

Este exemplo define o tempo limite do atraso de cancelamento de registro para o valor especificado para o grupo-alvo especificado.

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
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ModifyTargetGroupAttributes](#) na Referência de AWS CLI Comandos.

modify-target-group

O código de exemplo a seguir mostra como usar `modify-target-group`.

AWS CLI

Para modificar a configuração da verificação de integridade de um grupo-alvo

O `modify-target-group` exemplo a seguir altera a configuração das verificações de saúde usadas para avaliar a integridade dos alvos para o grupo-alvo especificado. Observe que, devido à forma como as vírgulas CLI são analisadas, você deve delimitar o intervalo da `--matcher` opção com aspas simples em vez de aspas duplas.

```
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'
```

Saída:

```
{  
  "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"
      }
    ]
  }

```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do Application Load Balancers.

- Para API obter detalhes, consulte [ModifyTargetGroup](#) na Referência de AWS CLI Comandos.

register-targets

O código de exemplo a seguir mostra como usar `register-targets`.

AWS CLI

Exemplo 1: Para registrar destinos com um grupo-alvo por ID de instância

O `register-targets` exemplo a seguir registra as instâncias especificadas com um grupo de destino. O grupo-alvo deve ter um tipo de alvo de `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

```

Exemplo 2: Para registrar alvos em um grupo-alvo usando substituições de porta

O `register-targets` exemplo a seguir registra a instância especificada com um grupo de destino usando várias portas. Isso permite que você registre contêineres na mesma instância que os destinos no grupo de destino.

```

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

```

Exemplo 3: Para registrar destinos com um grupo-alvo por endereço IP

O `register-targets` exemplo a seguir registra os endereços IP especificados com um grupo de destino. O grupo-alvo deve ter um tipo de alvo de `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
```

Exemplo 4: Para registrar uma função Lambda como alvo

O `register-targets` exemplo a seguir registra os endereços IP especificados com um grupo de destino. O grupo-alvo deve ter um tipo de alvo de Lambda. Você deve conceder permissão ao Elastic Load Balancing para invocar a função 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
```

- Para API obter detalhes, consulte [RegisterTargets](#) na Referência de AWS CLI Comandos.

remove-listener-certificates

O código de exemplo a seguir mostra como usar `remove-listener-certificates`.

AWS CLI

Para remover um certificado de um ouvinte seguro

Este exemplo remove o certificado especificado do ouvinte seguro especificado.

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
```

- Para API obter detalhes, consulte [RemoveListenerCertificates](#) na Referência de AWS CLI Comandos.

remove-tags

O código de exemplo a seguir mostra como usar `remove-tags`.

AWS CLI

Para remover tags de um balanceador de carga

O `remove-tags` exemplo a seguir remove as `department` tags `project` and do balanceador de carga especificado.

```
aws elbv2 remove-tags \  
  --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --tag-keys project department
```

- Para API obter detalhes, consulte [RemoveTags](#) na Referência de AWS CLI Comandos.

set-ip-address-type

O código de exemplo a seguir mostra como usar `set-ip-address-type`.

AWS CLI

Para definir o tipo de endereço de um balanceador de carga

Este exemplo define o tipo de endereço do balanceador de carga especificado como `dualstack`. As sub-redes do balanceador de carga devem ter blocos associados. 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
```

Saída:

```
{  
  "IpAddressType": "dualstack"  
}
```

- Para API obter detalhes, consulte [SetIpAddressType](#) na Referência de AWS CLI Comandos.

set-rule-priorities

O código de exemplo a seguir mostra como usar `set-rule-priorities`.

AWS CLI

Para definir a prioridade da regra

Este exemplo define a prioridade da regra especificada.

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
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [SetRulePriorities](#) na Referência de AWS CLI Comandos.

set-security-groups

O código de exemplo a seguir mostra como usar `set-security-groups`.

AWS CLI

Para associar um grupo de segurança a um balanceador de carga

Este exemplo associa o grupo de segurança especificado ao balanceador de carga especificado.

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
```

Saída:

```
{
  "SecurityGroupIds": [
    "sg-5943793c"
  ]
}
```

- Para API obter detalhes, consulte [SetSecurityGroups](#) na Referência de AWS CLI Comandos.

set-subnets

O código de exemplo a seguir mostra como usar `set-subnets`.

AWS CLI

Para habilitar zonas de disponibilidade para um balanceador de carga

Este exemplo ativa a Zona de Disponibilidade para a sub-rede especificada para o balanceador de carga especificado.

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
```

Saída:

```
{
  "AvailabilityZones": [
    {
      "SubnetId": "subnet-8360a9e7",
      "ZoneName": "us-west-2a"
    },
    {
      "SubnetId": "subnet-b7d581c0",
      "ZoneName": "us-west-2b"
    }
  ]
}
```

- Para API obter detalhes, consulte [SetSubnets](#) na Referência de AWS CLI Comandos.

Exemplos do Elastic Transcoder usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Elastic Transcoder.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-job

O código de exemplo a seguir mostra como usar `cancel-job`.

AWS CLI

Para cancelar um trabalho para ElasticTranscoder

Isso cancela o trabalho especificado para ElasticTranscoder.

Comando:

```
aws elastictranscoder cancel-job --id 333333333333-abcde3
```

- Para API obter detalhes, consulte [CancelJob](#) na Referência de AWS CLI Comandos.

create-job

O código de exemplo a seguir mostra como usar create-job.

AWS CLI

Para criar um emprego para ElasticTranscoder

O create-job exemplo a seguir cria um trabalho para 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
```

Conteúdo de inputs.json:

```
[{  
  "Key": "ETS_example_file.mp4",  
  "FrameRate": "auto",  
  "Resolution": "auto",  
  "AspectRatio": "auto",  
  "Interlaced": "auto",  
  "Container": "mp4"  
}]
```

Conteúdo de outputs.json:

```
[  
  {  
    "Key": "webm/ETS_example_file-kindlefirehd.webm",  
    "Rotate": "0",
```

```

    "PresetId": "1351620000001-100250"
  }
]

```

Conteúdo de `user-metadata.json`:

```

{
  "Food type": "Italian",
  "Cook book": "recipe notebook"
}

```

Saída:

```

{
  "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": "33333333333333-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"
  }
}

```

- Para API obter detalhes, consulte [CreateJob](#) na Referência de AWS CLI Comandos.

create-pipeline

O código de exemplo a seguir mostra como usar create-pipeline.

AWS CLI

Para criar um pipeline para ElasticTranscoder

O create-pipeline exemplo a seguir cria um pipeline para 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
```

Conteúdo de content-config.json:

```
{  
  "Bucket": "salesoffice.example.com-public-promos",  
  "Permissions": [  
    {  
      "GranteeType": "Email",  
      "Grantee": "marketing-promos@example.com",  
      "Access": [  
        "FullControl"  
      ]  
    }  
  ],  
  "StorageClass": "Standard"  
}
```

Conteúdo de thumbnail-config.json:

```
{  
  "Bucket": "salesoffice.example.com-public-promos-thumbnails",  
  "Permissions": [  
    {  
      "GranteeType": "Email",  
      "Grantee": "marketing-promos@example.com",  
      "Access": [  
        "FullControl"  
      ]  
    }  
  ],  
  "StorageClass": "ReducedRedundancy"  
}
```

Saída:

```
{  
  "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"
      }
    ]
  }

```

- Para API obter detalhes, consulte [CreatePipeline](#) na Referência de AWS CLI Comandos.

create-preset

O código de exemplo a seguir mostra como usar create-preset.

AWS CLI

Para criar uma predefinição para ElasticTranscoder

O create-preset exemplo a seguir cria uma predefinição para 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

```

Conteúdo de 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"
      }
    ]
  }
}

```

Conteúdo de audio.json:

```

{
  "Codec": "AAC",
  "CodecOptions": {
    "Profile": "AAC-LC"
  },
  "SampleRate": "44100",
  "BitRate": "96",
  "Channels": "2"
}

```

Conteúdo de thumbnails.json:

```

{
  "Format": "png",
  "Interval": "120",
  "MaxWidth": "auto",
  "MaxHeight": "auto",
  "SizingPolicy": "Fit",
  "PaddingPolicy": "Pad"
}

```

```
}
```

Saída:

```
{
  "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": ""
}

```

- Para API obter detalhes, consulte [CreatePreset](#) na Referência de AWS CLI Comandos.

delete-pipeline

O código de exemplo a seguir mostra como usar delete-pipeline.

AWS CLI

Para excluir o ElasticTranscoder pipeline especificado

Isso exclui o ElasticTranscoder pipeline especificado.

Comando:

```
aws elastictranscoder delete-pipeline --id 111111111111-abcde1
```

Saída:

```
{
  "Success": "true"
}
```

- Para API obter detalhes, consulte [DeletePipeline](#) na Referência de AWS CLI Comandos.

delete-preset

O código de exemplo a seguir mostra como usar `delete-preset`.

AWS CLI

Para excluir a ElasticTranscoder predefinição especificada

Isso exclui a ElasticTranscoder predefinição especificada.

Comando:

```
aws elastictranscoder delete-preset --id 555555555555-abcde5
```

- Para API obter detalhes, consulte [DeletePreset](#) na Referência de AWS CLI Comandos.

list-jobs-by-pipeline

O código de exemplo a seguir mostra como usar `list-jobs-by-pipeline`.

AWS CLI

Para recuperar uma lista de ElasticTranscoder trabalhos no pipeline especificado

Este exemplo recupera uma lista de ElasticTranscoder trabalhos no pipeline especificado.

Comando:

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 111111111111-abcde1
```

Saída:

```
{
  "Jobs": []
}
```

- Para API obter detalhes, consulte [ListJobsByPipeline](#) na Referência de AWS CLI Comandos.

list-jobs-by-status

O código de exemplo a seguir mostra como usar `list-jobs-by-status`.

AWS CLI

Para recuperar uma lista de ElasticTranscoder trabalhos com o status Concluído

Este exemplo recupera uma lista de ElasticTranscoder trabalhos com o status Concluído.

Comando:

```
aws elastictranscoder list-jobs-by-status --status Complete
```

Saída:

```
{
  "Jobs": []
}
```

- Para API obter detalhes, consulte [ListJobsByStatus](#) na Referência de AWS CLI Comandos.

list-pipelines

O código de exemplo a seguir mostra como usar `list-pipelines`.

AWS CLI

Para recuperar uma lista de pipelines ElasticTranscoder

Este exemplo recupera uma lista de ElasticTranscoder pipelines.

Comando:

aws elastictranscoder list-pipelines

Saída:

```
{
  "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": "333333333333-abcde3",
      "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:pipeline/333333333333-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": "333333333333-abcde2",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-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": "333333333333-abcde1",
    "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde1"
}
]
}

```

- Para API obter detalhes, consulte [ListPipelines](#) na Referência de AWS CLI Comandos.

list-presets

O código de exemplo a seguir mostra como usar `list-presets`.

AWS CLI

Para recuperar uma lista de predefinições ElasticTranscoder

Este exemplo recupera uma lista de ElasticTranscoder predefinições.

Comando:

```
aws elastictranscoder list-presets --max-items 2
```

Saída:

```
{
  "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=="
}

```

- Para API obter detalhes, consulte [ListPresets](#) na Referência de AWS CLI Comandos.

read-job

O código de exemplo a seguir mostra como usar `read-job`.

AWS CLI

Para recuperar um trabalho ElasticTranscoder

Este exemplo recupera o ElasticTranscoder trabalho especificado.

Comando:

```
aws elastictranscoder read-job --id 1533838012294-example
```

Saída:

```

{
  "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"
}
}
```

- Para API obter detalhes, consulte [ReadJob](#) na Referência de AWS CLI Comandos.

read-pipeline

O código de exemplo a seguir mostra como usar `read-pipeline`.

AWS CLI

Para recuperar um pipeline ElasticTranscoder

Este exemplo recupera o ElasticTranscoder pipeline especificado.

Comando:

```
aws elastictranscoder read-pipeline --id 333333333333-abcde3
```

Saída:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    },
    "Name": "Default",
    "ThumbnailConfig": {
      "Bucket": "ets-example",
      "StorageClass": "ReducedRedundancy",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    }
  }
}
```

```

    }
  ]
},
"Notifications": {
  "Completed": "",
  "Warning": "",
  "Progressing": "",
  "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
},
"Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
"InputBucket": "ets-example",
"Id": "333333333333-abcde3",
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde3"
},
"Warnings": [
  {
    "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
    "Code": "6006"
  }
]
}

```

- Para API obter detalhes, consulte [ReadPipeline](#) na Referência de AWS CLI Comandos.

read-preset

O código de exemplo a seguir mostra como usar read-preset.

AWS CLI

Para recuperar uma predefinição ElasticTranscoder

Este exemplo recupera a ElasticTranscoder predefinição especificada.

Comando:

```
aws elastictranscoder read-preset --id 1351620000001-500020
```


Saída:

```
{
  "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"
  }
}

```

- Para API obter detalhes, consulte [ReadPreset](#) na Referência de AWS CLI Comandos.

update-pipeline-notifications

O código de exemplo a seguir mostra como usar `update-pipeline-notifications`.

AWS CLI

Para atualizar as notificações de um ElasticTranscoder pipeline

Este exemplo atualiza as notificações do ElasticTranscoder pipeline especificado.

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

```

Saída:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [UpdatePipelineNotifications](#) na Referência de AWS CLI Comandos.

update-pipeline-status

O código de exemplo a seguir mostra como usar update-pipeline-status.

AWS CLI

Para atualizar o status de um ElasticTranscoder pipeline

Este exemplo atualiza o status do ElasticTranscoder pipeline especificado.

Comando:

```
aws elastictranscoder update-pipeline-status --id 111111111111-abcde1 --  
status Paused
```

Saída:

```
{  
  "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"
  }
}

```

- Para API obter detalhes, consulte [UpdatePipelineStatus](#) na Referência de AWS CLI Comandos.

update-pipeline

O código de exemplo a seguir mostra como usar `update-pipeline`.

AWS CLI

Para atualizar um ElasticTranscoder pipeline

O `update-pipeline` exemplo a seguir atualiza o ElasticTranscoder pipeline especificado.

```

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

```

Conteúdo de `content-config.json`:

```

{
  "Bucket": "salesoffice.example.com-public-promos",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",
      "Access": [
        "FullControl"
      ]
    }
  ],
  "StorageClass": "Standard"
}

```

```
}
```

Conteúdo de thumbnail-config.json:

```
{
  "Bucket": "salesoffice.example.com-public-promos-thumbnails",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",
      "Access": [
        "FullControl"
      ]
    }
  ],
  "StorageClass": "ReducedRedundancy"
}
```

Saída:

```
{
  "Pipeline": {
    "Status": "Active",
    "ContentConfig": {
      "Bucket": "ets-example",
      "StorageClass": "Standard",
      "Permissions": [
        {
          "Access": [
            "FullControl"
          ],
          "Grantee": "marketing-promos@example.com",
          "GranteeType": "Email"
        }
      ]
    },
    "Name": "DefaultExample",
    "ThumbnailConfig": {
      "Bucket": "ets-example",
      "StorageClass": "ReducedRedundancy",
      "Permissions": [
        {
          "Access": [
```

```

        "FullControl"
      ],
      "Grantee": "marketing-promos@example.com",
      "GranteeType": "Email"
    }
  ]
},
"Notifications": {
  "Completed": "",
  "Warning": "",
  "Progressing": "",
  "Error": "arn:aws:sns:us-east-1:111222333444:ETS_Errors"
},
"Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
"InputBucket": "ets-example",
"Id": "333333333333-abcde3",
"Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/333333333333-abcde3"
},
"Warnings": [
  {
    "Message": "The SNS notification topic for Error events and the pipeline
are in different regions, which increases processing time for jobs in the pipeline
and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
pipeline.",
    "Code": "6006"
  }
]
}

```

- Para API obter detalhes, consulte [UpdatePipeline](#) na Referência de AWS CLI Comandos.

ElastiCache exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with ElastiCache.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-resource

O código de exemplo a seguir mostra como usar `add-tags-to-resource`.

AWS CLI

Para adicionar tags a um recurso

O `add-tags-to-resource` exemplo a seguir adiciona até 10 tags, pares de valores-chave, a um cluster ou recurso de 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"}
```

Saída:

```
{  
  "TagList": [  
    {  
      "Value": "20150202",  
      "Key": "APIVersion"  
    },  
    {  
      "Value": "ElastiCache",  
      "Key": "Service"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Monitoramento de custos com tags de alocação de custos no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [AddTagsToResource](#) na Referência de AWS CLI Comandos.

authorize-cache-security-group-ingress

O código de exemplo a seguir mostra como usar `authorize-cache-security-group-ingress`.

AWS CLI

Para autorizar a entrada do grupo de segurança de cache

O `authorize-cache-security-group-ingress` exemplo a seguir permite a entrada de rede em um grupo de segurança de 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"
```

O comando não produz nenhuma saída.

Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [AuthorizeCacheSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

batch-apply-update-action

O código de exemplo a seguir mostra como usar `batch-apply-update-action`.

AWS CLI

Para aplicar uma atualização de serviço

O `batch-apply-update-action` exemplo a seguir aplica uma atualização de serviço a um cluster Redis.

```
aws elasticache batch-apply-update-action \  
  --service-update-name elc-xxxxx406-xxx \  
  --replication-group-ids test-cluster
```

Saída:

```
{
  "ProcessedUpdateActions": [
    {
      "ReplicationGroupId": "pat-cluster",
      "ServiceUpdateName": "elc-xxxxx406-xxx",
      "UpdateActionStatus": "waiting-to-start"
    }
  ],
  "UnprocessedUpdateActions": []
}
```

Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [BatchApplyUpdateAction](#) na Referência de AWS CLI Comandos.

batch-stop-update-action

O código de exemplo a seguir mostra como usar batch-stop-update-action.

AWS CLI

Para interromper uma atualização de serviço

O batch-stop-update-action exemplo a seguir aplica uma atualização de serviço a um cluster Redis.

```
aws elasticache batch-stop-update-action \
  --service-update-name elc-xxxxx406-xxx \
  --replication-group-ids test-cluster
```

Saída:

```
{
  "ProcessedUpdateActions": [
    {
      "ReplicationGroupId": "pat-cluster",
      "ServiceUpdateName": "elc-xxxxx406-xxx",
```

```

        "UpdateActionStatus": "stopping"
    }
],
"UnprocessedUpdateActions": []
}

```

Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache no Guia](#) do usuário do Elasticache.

- Para API obter detalhes, consulte [BatchStopUpdateAction](#) na Referência de AWS CLI Comandos.

copy-snapshot

O código de exemplo a seguir mostra como usar copy-snapshot.

AWS CLI

Para copiar um instantâneo

O copy-snapshot exemplo a seguir faz uma cópia de um snapshot existente.

```

aws elasticache copy-snapshot \
  --source-snapshot-name "my-snapshot" \
  --target-snapshot-name "my-snapshot-copy"

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Exportação de um Backup no Guia](#) do Usuário do ElastiCache.

- Para API obter detalhes, consulte [CopySnapshot](#) na Referência de AWS CLI Comandos.

create-cache-cluster

O código de exemplo a seguir mostra como usar `create-cache-cluster`.

AWS CLI

Para criar um cluster de cache

O `create-cache-cluster` exemplo a seguir cria um cluster de cache usando o mecanismo Redis.

```

aws elasticache create-cache-cluster \
  --cache-cluster-id "cluster-test" \
  --engine redis \
  --cache-node-type cache.m5.large \
  --num-cache-nodes 1

```

Saída:

```
{
```

```
"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
}
}
```

Para obter mais informações, consulte [Criação de um cluster](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [CreateCacheCluster](#) na Referência de AWS CLI Comandos.

create-cache-parameter-group

O código de exemplo a seguir mostra como usar create-cache-parameter-group.

AWS CLI

Para criar um grupo de parâmetros de cache

O create-cache-parameter-group exemplo a seguir cria um novo grupo de parâmetros de ElastiCache cache da Amazon.

```
aws elasticache create-cache-parameter-group \
  --cache-parameter-group-family "redis5.0" \
```

```
--cache-parameter-group-name "mygroup" \  
--description "mygroup"
```

Saída:

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "mygroup",  
    "CacheParameterGroupFamily": "redis5.0",  
    "Description": "my group"  
  }  
}
```

Para obter mais informações, consulte [Criação de um grupo de parâmetros](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [CreateCacheParameterGroup](#) na Referência de AWS CLI Comandos.

create-cache-subnet-group

O código de exemplo a seguir mostra como usar create-cache-subnet-group.

AWS CLI

Para criar um grupo de sub-redes de cache

O create-cache-subnet-group exemplo a seguir cria um novo grupo de sub-redes de cache.

```
aws elasticache create-cache-subnet-group \  
--cache-subnet-group-name "mygroup" \  
--cache-subnet-group-description "my subnet group" \  
--subnet-ids "subnet-xxxxec4f"
```

Saída:

```
{  
  "CacheSubnetGroup": {  
    "CacheSubnetGroupName": "mygroup",  
    "CacheSubnetGroupDescription": "my subnet group",  
    "VpcId": "vpc-a3e97cdb",  
  }  
}
```

```

    "Subnets": [
      {
        "SubnetIdentifier": "subnet-xxxxec4f",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2d"
        }
      }
    ]
  }
}

```

Para obter mais informações, consulte [Criação de um grupo de sub-rede de cache no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [CreateCacheSubnetGroup](#) na Referência de AWS CLI Comandos.

create-global-replication-group

O código de exemplo a seguir mostra como usar `create-global-replication-group`.

AWS CLI

Para criar um grupo de replicação global

O `create-global-replication-group` exemplo a seguir cria um novo grupo de replicação global.

```

aws elasticache create-global-replication-group \
  --global-replication-group-id-suffix my-global-replication-group \
  --primary-replication-group-id my-primary-cluster

```

Saída:

```

{
  "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": [
      {
        "GlobalNodeId": "sgaui-my-global-replication-group-0001",
        "Slots": "0-16383"
      }
    ],
    "AuthTokenEnabled": false,
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}

```

Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global no Guia do usuário](#) do ElastiCache.

- Para API obter detalhes, consulte [CreateGlobalReplicationGroup](#) na Referência de AWS CLI Comandos.

create-replication-group

O código de exemplo a seguir mostra como usar `create-replication-group`.

AWS CLI

Para criar um grupo de replicação

O `create-replication-group` exemplo a seguir cria um grupo de replicação Redis (modo de cluster desativado) ou Redis (modo de cluster ativado). Essa operação é válida somente para o Redis.

```

aws elasticache create-replication-group \
  --replication-group-id "mygroup" \

```

```
--replication-group-description "my group" \  
--engine "redis" \  
--cache-node-type "cache.m5.large"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Criação de um grupo de replicação do Redis](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [CreateReplicationGroup](#) na Referência de AWS CLI Comandos.

create-snapshot

O código de exemplo a seguir mostra como usar create-snapshot.

AWS CLI

Para criar um instantâneo

O create-snapshot exemplo a seguir cria um instantâneo usando o mecanismo Redis.

```
aws elasticache create-snapshot \  

```

```
--snapshot-name mysnapshot \  
--cache-cluster-id cluster-test
```

Saída:

```
{  
  "Snapshot": {  
    "SnapshotName": "mysnapshot",  
    "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Backup e restauração ElastiCache para Redis](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [CreateSnapshot](#) na Referência de AWS CLI Comandos.

create-user-group

O código de exemplo a seguir mostra como usar `create-user-group`.

AWS CLI

Para criar um grupo de usuários

O `create-user-group` exemplo a seguir cria um novo grupo de usuários.

```
aws elasticache create-user-group \  
  --user-group-id myusergroup \  
  --engine redis \  
  --user-ids default
```

Saída:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "creating",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"  
}
```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [CreateUserGroup](#) na Referência de AWS CLI Comandos.

create-user

O código de exemplo a seguir mostra como usar `create-user`.

AWS CLI

Para criar um usuário

O `create-user` exemplo a seguir cria um novo usuário.

```
aws elasticache create-user \  
  --user-id user1 \  
  --user-name myUser \  
  --user-arn arn:aws:iam::123456789012:user/myUser
```

```
--passwords mYnuUzrpAxXw2rdzx \
--engine redis \
--access-string "on ~app:* -@all +@read"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

decrease-node-groups-in-global-replication-group

O código de exemplo a seguir mostra como usar `decrease-node-groups-in-global-replication-group`.

AWS CLI

Para diminuir o número de grupos de nós em um grupo de replicação global

O seguinte `decrease-node-groups-in-global-replication-group` diminui a contagem de grupos de nós usando o mecanismo Redis.

```
aws elasticache decrease-node-groups-in-global-replication-group \
--global-replication-group-id sgau-test \
--node-group-count 1 \
```

```
--apply-immediately \  
--global-node-groups-to-retain sgai-test-0003
```

Saída:

```
{  
  "GlobalReplicationGroup":  
  {  
    "GlobalReplicationGroupId": "sgai-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": "sgai-test-0001",  
        "Slots": "0-449,1816-5461"  
      },  
      {  
        "GlobalNodeGroupId": "sgai-test-0002",  
        "Slots": "6827-10922"  
      },  
      {  
        "GlobalNodeGroupId": "sgai-test-0003",  
        "Slots": "10923-14052,15418-16383"  
      }  
    ]  
  }  
}
```

```

    },
    {
      "GlobalNodeId": "sgai-test-0004",
      "Slots": "450-1815,5462-6826,14053-15417"
    }
  ],
  "AuthTokenEnabled": false,
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
}

```

Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global no Guia do usuário](#) do Elasticache.

- Para API obter detalhes, consulte [DecreaseNodeGroupsInGlobalReplicationGroup](#) na Referência de AWS CLI Comandos.

decrease-replica-count

O código de exemplo a seguir mostra como usar decrease-replica-count.

AWS CLI

Para diminuir a contagem de réplicas

O decrease-replica-count exemplo a seguir diminui dinamicamente o número de réplicas em um grupo de replicação Redis (modo de cluster desativado) ou o número de nós de réplica em um ou mais grupos de nós (fragmentos) de um grupo de replicação Redis (modo de cluster ativado). Essa operação é executada sem tempo de inatividade do cluster.

```

aws elasticache decrease-replica-count \
  --replication-group-id my-cluster \
  --apply-immediately \
  --new-replica-count 2

```

Saída:

```

{
  "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
}
}

```

Para obter mais informações, consulte [Alterando o número de réplicas no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [DecreaseReplicaCount](#)na Referência de AWS CLI Comandos.

delete-cache-cluster

O código de exemplo a seguir mostra como usar `delete-cache-cluster`.

AWS CLI

Para excluir um cluster de cache

O `delete-cache-cluster` exemplo a seguir exclui o cluster especificado previamente provisionado. O comando exclui todos os nós de cache associados, os pontos finais dos nós e o próprio cluster. Quando você recebe uma resposta bem-sucedida dessa operação, a Amazon ElastiCache imediatamente começa a excluir o cluster; você não pode cancelar ou reverter essa operação.

Essa operação não é válida para o seguinte:

Clusters Redis (modo de cluster ativado) Um cluster que é a última réplica de leitura de um grupo de replicaçãoUm grupo de nós (fragmento) que tem o modo Multi-AZ ativadoUm cluster de um grupo de replicação Redis (modo de cluster ativado) Um cluster que não está no estado disponível

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id "my-cluster-002"
```

Saída:

```
{  
  "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
}
}

```

Para obter mais informações, consulte [Excluindo um cluster](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DeleteCacheCluster](#) na Referência de AWS CLI Comandos.

delete-cache-parameter-group

O código de exemplo a seguir mostra como usar `delete-cache-parameter-group`.

AWS CLI

Para excluir um grupo de parâmetros de cache

O `delete-cache-parameter-group` exemplo a seguir exclui o grupo de parâmetros de cache especificado. Você não pode excluir um grupo de parâmetros de cache se ele estiver associado a qualquer cluster de cache.

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myparamgroup
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um grupo de parâmetros no Guia](#) do usuário do Elasticache.

- Para API obter detalhes, consulte [DeleteCacheParameterGroup](#) na Referência de AWS CLI Comandos.

delete-cache-subnet-group

O código de exemplo a seguir mostra como usar `delete-cache-subnet-group`.

AWS CLI

Para excluir um grupo de sub-redes de cache

O `delete-cache-subnet-group` exemplo a seguir exclui o grupo de sub-rede de cache especificado. Você não pode excluir um grupo de sub-rede de cache se ele estiver associado a algum cluster.

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name "mygroup"
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um grupo de sub-redes](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [DeleteCacheSubnetGroup](#) na Referência de AWS CLI Comandos.

delete-global-replication-group

O código de exemplo a seguir mostra como usar `delete-global-replication-group`.

AWS CLI

Para excluir um grupo de replicação global

O `delete-global-replication-group` exemplo a seguir exclui um novo grupo de replicação global.

```
aws elasticache delete-global-replication-group \
  --global-replication-group-id my-global-replication-group \
  --retain-primary-replication-group
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global no Guia do usuário](#) do Elasticache.

- Para API obter detalhes, consulte [DeleteGlobalReplicationGroup](#) na Referência de AWS CLI Comandos.

delete-replication-group

O código de exemplo a seguir mostra como usar `delete-replication-group`.

AWS CLI

Para excluir um grupo de replicação

O `delete-replication-group` exemplo a seguir exclui um grupo de replicação existente. Por padrão, essa operação exclui todo o grupo de replicação, incluindo o primário/primário e todas as réplicas de leitura. Se o grupo de replicação tiver somente um primário, você poderá, opcionalmente, excluir somente as réplicas de leitura, mantendo o primário definindo `=true`. `RetainPrimaryCluster`

Quando você recebe uma resposta bem-sucedida dessa operação, a Amazon ElastiCache imediatamente começa a excluir os recursos selecionados; você não pode cancelar ou reverter essa operação. Válido somente para Redis.

```
aws elasticache delete-replication-group \  
  --replication-group-id "mygroup"
```

Saída:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mygroup",  
    "Description": "my group",  
    "Status": "deleting",  
    "PendingModifiedValues": {},  
    "AutomaticFailover": "disabled",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:00-07:00",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

- Para API obter detalhes, consulte [DeleteReplicationGroup](#) na Referência de AWS CLI Comandos.

delete-snapshot

O código de exemplo a seguir mostra como usar `delete-snapshot`.

AWS CLI

Para excluir um snapshot

O `delete-snapshot` exemplo a seguir excluiu um instantâneo usando o mecanismo Redis.

```
aws elasticache delete-snapshot \  
  --snapshot-name mysnapshot
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Backup e restauração ElastiCache para Redis](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DeleteSnapshot](#) na Referência de AWS CLI Comandos.

delete-user-group

O código de exemplo a seguir mostra como usar delete-user-group.

AWS CLI

Para excluir um grupo de usuários

O delete-user-group exemplo a seguir exclui um grupo de usuários.

```

aws elasticache delete-user-group \
  --user-group-id myusergroup

```


Saída:

```
{
  "UserGroupId": "myusergroup",
  "Status": "deleting",
  "Engine": "redis",
  "UserIds": [
    "default"
  ],
  "ReplicationGroups": [],
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [DeleteUserGroup](#) na Referência de AWS CLI Comandos.

delete-user

O código de exemplo a seguir mostra como usar delete-user.

AWS CLI

Como excluir um usuário

O delete-user exemplo a seguir exclui um usuário.

```
aws elasticache delete-user \
  --user-id user2
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

describe-cache-clusters

O código de exemplo a seguir mostra como usar `describe-cache-clusters`.

AWS CLI

Para descrever um cluster de cache

O `describe-cache-clusters` exemplo a seguir descreve um cluster de cache.

```
aws elasticache describe-cache-clusters
```

Saída:

```
{
  "CacheClusters": [
    {
      "CacheClusterId": "my-cluster-003",
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.5",
      "CacheClusterStatus": "available",
      "NumCacheNodes": 1,
      "PreferredAvailabilityZone": "us-west-2a",
      "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
      "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
      "PendingModifiedValues": {},
      "NotificationConfiguration": {
        "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxxx152:My_Topic",
        "TopicStatus": "active"
      }
    }
  ]
}
```

```

    },
    "CacheSecurityGroups": [],
    "CacheParameterGroup": {
        "CacheParameterGroupName": "default.redis5.0",
        "ParameterApplyStatus": "in-sync",
        "CacheNodeIdsToReboot": []
    },
    "CacheSubnetGroupName": "kxkxk",
    "AutoMinorVersionUpgrade": true,
    "SecurityGroups": [
        {
            "SecurityGroupId": "sg-xxxxxd7b",
            "Status": "active"
        }
    ],
    "ReplicationGroupId": "my-cluster",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:30-07:30",
    "AuthTokenEnabled": false,
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false,
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx152:cluster:my-cache-
cluster",
    "ReplicationGroupLogDeliveryEnabled": false,
    "LogDeliveryConfigurations": [
        {
            "LogType": "slow-log",
            "DestinationType": "cloudwatch-logs",
            "DestinationDetails": {
                "CloudWatchLogsDetails": {
                    "LogGroup": "test-log"
                }
            },
            "LogFormat": "text",
            "Status": "active"
        }
    ]
}
]
}
}

```

Para obter mais informações, consulte [Gerenciamento de clusters](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DescribeCacheClusters](#) na Referência de AWS CLI Comandos.

describe-cache-engine-versions

O código de exemplo a seguir mostra como usar `describe-cache-engine-versions`.

AWS CLI

Para descrever uma versão do mecanismo de cache

O `describe-cache-engine-versions` exemplo a seguir retorna uma lista dos mecanismos de cache disponíveis e suas versões.

```
aws elasticache describe-cache-engine-versions \  
  --engine "Redis"
```

Saída:

```
{  
  "CacheEngineVersions": [  
    {  
      "Engine": "redis",  
      "EngineVersion": "2.6.13",  
      "CacheParameterGroupFamily": "redis2.6",  
      "CacheEngineDescription": "Redis",  
      "CacheEngineVersionDescription": "redis version 2.6.13"  
    },  
    {  
      "Engine": "redis",  
      "EngineVersion": "2.8.19",  
      "CacheParameterGroupFamily": "redis2.8",  
      "CacheEngineDescription": "Redis",  
      "CacheEngineVersionDescription": "redis version 2.8.19"  
    },  
    {  
      "Engine": "redis",  
      "EngineVersion": "2.8.21",  
      "CacheParameterGroupFamily": "redis2.8",  
      "CacheEngineDescription": "Redis",  
      "CacheEngineVersionDescription": "redis version 2.8.21"  
    }  
  ]  
}
```

```
{
  "Engine": "redis",
  "EngineVersion": "2.8.22",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.22"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.23",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.23"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.24",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.24"
},
{
  "Engine": "redis",
  "EngineVersion": "2.8.6",
  "CacheParameterGroupFamily": "redis2.8",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 2.8.6"
},
{
  "Engine": "redis",
  "EngineVersion": "3.2.10",
  "CacheParameterGroupFamily": "redis3.2",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 3.2.10"
},
{
  "Engine": "redis",
  "EngineVersion": "3.2.4",
  "CacheParameterGroupFamily": "redis3.2",
  "CacheEngineDescription": "Redis",
  "CacheEngineVersionDescription": "redis version 3.2.4"
},
{
  "Engine": "redis",
```

```
    "EngineVersion": "3.2.6",
    "CacheParameterGroupFamily": "redis3.2",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 3.2.6"
  },
  {
    "Engine": "redis",
    "EngineVersion": "4.0.10",
    "CacheParameterGroupFamily": "redis4.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 4.0.10"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.0",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.0"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.3",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.3"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.4",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.4"
  },
  {
    "Engine": "redis",
    "EngineVersion": "5.0.5",
    "CacheParameterGroupFamily": "redis5.0",
    "CacheEngineDescription": "Redis",
    "CacheEngineVersionDescription": "redis version 5.0.5"
  }
]
}
```

- Para API obter detalhes, consulte [DescribeCacheEngineVersions](#) na Referência de AWS CLI Comandos.

describe-cache-parameter-groups

O código de exemplo a seguir mostra como usar `describe-cache-parameter-groups`.

AWS CLI

Para descrever um grupo de parâmetros de cache

O `describe-cache-parameter-groups` exemplo a seguir retorna uma lista de descrições de grupos de parâmetros de cache.

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name "mygroup"
```

Saída:

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "mygroup",
      "CacheParameterGroupFamily": "redis5.0",
      "Description": " "
    }
  ]
}
```

Para obter mais informações, consulte [Configurando parâmetros do mecanismo usando grupos de parâmetros no Guia](#) do usuário do Elasticache.

- Para API obter detalhes, consulte [DescribeCacheParameterGroups](#) na Referência de AWS CLI Comandos.

describe-cache-parameters

O código de exemplo a seguir mostra como usar `describe-cache-parameters`.

AWS CLI

Para descrever os parâmetros do cache

O exemplo "describe-cache-parameters" a seguir retorna a lista detalhada de parâmetros para o grupo de parâmetros de cache especificado.

```
aws elasticache describe-cache-parameters \  
--cache-parameter-group-name "myparamgroup"
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Gerenciamento de parâmetros](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DescribeCacheParameters](#) na Referência de AWS CLI Comandos.

describe-cache-subnet-groups

O código de exemplo a seguir mostra como usar `describe-cache-subnet-groups`.

AWS CLI

Para descrever grupos de sub-redes de cache

O `describe-cache-subnet-groups` exemplo a seguir retorna uma lista de grupos de sub-redes.

```
aws elasticache describe-cache-subnet-groups
```

Saída:

```
{
  "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"
          }
        }
      ]
    }
  ]
}

```

Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#) no Guia do usuário do ElastiCache ou Sub-redes e grupos de sub-redes no Guia do usuário do [Memcached](#). ElastiCache

- Para API obter detalhes, consulte [DescribeCacheSubnetGroups](#) na Referência de AWS CLI Comandos.

describe-engine-default-parameters

O código de exemplo a seguir mostra como usar `describe-engine-default-parameters`.

AWS CLI

Para descrever os parâmetros padrão do motor

O `describe-engine-default-parameters` exemplo a seguir retorna as informações padrão do mecanismo e dos parâmetros do sistema para o mecanismo de cache especificado.

```

aws elasticache describe-engine-default-parameters \
  --cache-parameter-group-family redis5.0

```

Saída:

```

{
  "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"
    }
]
}
}

```

- Para API obter detalhes, consulte [DescribeEngineDefaultParameters](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para descrever eventos de um grupo de replicação

O `describe-events` exemplo a seguir retorna uma lista de eventos para um grupo de replicação.

```

aws elasticache describe-events \
  --source-identifier test-cluster \
  --source-type replication-group

```

Saída:

```

{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Monitoramento de eventos](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-global-replication-groups

O código de exemplo a seguir mostra como usar `describe-global-replication-groups`.

AWS CLI

Para descrever grupos de replicação global

O `describe-global-replication-groups` exemplo a seguir retorna detalhes de um armazenamento de dados global.

```
aws elasticache describe-global-replication-groups \
  --global-replication-group-id my-grg
```

Saída:

```
{
  "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
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global no Guia do usuário](#) do ElastiCache.

- Para API obter detalhes, consulte [DescribeGlobalReplicationGroups](#) na Referência de AWS CLI Comandos.

describe-replication-groups

O código de exemplo a seguir mostra como usar `describe-replication-groups`.

AWS CLI

Para retornar uma lista de detalhes do grupo de replicação

O `describe-replication-groups` exemplo a seguir retorna os grupos de replicação.

```
aws elasticache describe-replication-groups
```

Saída:

```
{
  "ReplicationGroups": [
    {
      "ReplicationGroupId": "my-cluster",
      "Description": "mycluster",
      "Status": "available",
      "PendingModifiedValues": {},
      "MemberClusters": [
        "pat-cluster-001",
        "pat-cluster-002",
        "pat-cluster-003",
        "pat-cluster-004"
      ],
      "NodeGroups": [
        {
          "NodeGroupId": "0001",
          "Status": "available",
          "PrimaryEndpoint": {
            "Address": "my-
cluster.xxxxih.ng.0001.usw2.cache.amazonaws.com",
            "Port": 6379
          }
        }
      ]
    }
  ]
}
```

```
    },
    "ReaderEndpoint": {
      "Address": "my-cluster-
ro.xxxxih.ng.0001.usw2.cache.amazonaws.com",
      "Port": 6379
    },
    "NodeGroupMembers": [
      {
        "CacheClusterId": "my-cluster-001",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "pat-
cluster-001.xxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "primary"
      },
      {
        "CacheClusterId": "my-cluster-002",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "pat-
cluster-002.xxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "replica"
      },
      {
        "CacheClusterId": "my-cluster-003",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Address": "pat-
cluster-003.xxxxih.0001.usw2.cache.amazonaws.com",
          "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "replica"
      },
      {
        "CacheClusterId": "my-cluster-004",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
```

```

        "Address": "pat-
cluster-004.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:xxxxxxxxxxxx152:replicationgroup:my-cluster",
"LogDeliveryConfigurations": [
    {
        "LogType": "slow-log",
        "DestinationType": "cloudwatch-logs",
        "DestinationDetails": {
            "CloudWatchLogsDetails": {
                "LogGroup": "test-log"
            }
        },
        "LogFormat": "json",
        "Status": "active"
    }
]
}
]
}
}

```

Para obter mais informações, consulte [Gerenciamento de clusters](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DescribeReplicationGroups](#) na Referência de AWS CLI Comandos.

describe-reserved-cache-nodes-offerings

O código de exemplo a seguir mostra como usar `describe-reserved-cache-nodes-offerings`.

AWS CLI

Para descrever `reserved-cache-nodes-offerings`

O `describe-reserved-cache-nodes-offerings` exemplo a seguir retorna detalhes de uma `reserved-cache-node` opção.

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Saída:

```
{
  "ReservedCacheNodesOfferings": [
    {
      "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eacbc8e5",
      "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"
      }
    ]
  },
  ...
]
}

```

Para obter mais informações, consulte [Obter informações sobre as ofertas de nós reservados](#) no Guia do usuário do Elasticache Redis ou [Obter informações sobre as ofertas de nós reservados](#) no Guia do usuário do Elasticache Memcached.

- Para API obter detalhes, consulte [DescribeReservedCacheNodesOfferings](#) na Referência de AWS CLI Comandos.

describe-reserved-cache-nodes

O código de exemplo a seguir mostra como usar `describe-reserved-cache-nodes`.

AWS CLI

Para descrever os nós de cache reservados

O `describe-reserved-cache-nodes` exemplo a seguir retorna informações sobre nós de cache reservados para essa conta ou sobre o nó de cache reservado especificado.

elasticache da mandíbula `describe-reserved-cache-nodes`

Saída:

```

{
  "ReservedCacheNodes": [
    {
      "ReservedCacheNodeId": "mynode",
      "ReservedCacheNodesOfferingId": "xxxxxxxxxx-xxxxxx-xxxxxx-xxxxx-xxxxxxxxxx71",
      "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:xxxxxxxxxxxxx52:reserved-instance:mynode"
  }
]
}

```

Para obter mais informações, consulte [Gerenciamento de custos com nós reservados no Guia](#) do usuário do Elasticache.

- Para API obter detalhes, consulte [DescribeReservedCacheNodes](#) na Referência de AWS CLI Comandos.

describe-service-updates

O código de exemplo a seguir mostra como usar describe-service-updates.

AWS CLI

Para descrever as atualizações do serviço

O describe-service-updates exemplo a seguir retorna detalhes sobre atualizações de serviços.

```
aws elasticache describe-service-updates
```

Saída:

```

{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [DescribeServiceUpdates](#) na Referência de AWS CLI Comandos.

describe-snapshots

O código de exemplo a seguir mostra como usar `describe-snapshots`.

AWS CLI

Para descrever instantâneos

O exemplo de “`describe-snapshots`” a seguir retorna informações sobre os instantâneos do cluster ou do grupo de replicação.

aws elasticache describe-snapshots

Saída:

```
{
  "Snapshots": [
    {
      "SnapshotName": "automatic.my-cluster2-002-2019-12-05-06-38",
      "CacheClusterId": "my-cluster2-002",
      "SnapshotStatus": "available",
      "SnapshotSource": "automated",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.5",
      "NumCacheNodes": 1,
      "PreferredAvailabilityZone": "us-west-2a",
      "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
      "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
      "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxx52:My_Topic",
      "Port": 6379,
      "CacheParameterGroupName": "default.redis5.0",
      "CacheSubnetGroupName": "kxkxk",
      "VpcId": "vpc-a3e97cdb",
      "AutoMinorVersionUpgrade": true,
      "SnapshotRetentionLimit": 1,
      "SnapshotWindow": "06:30-07:30",
      "NodeSnapshots": [
        {
          "CacheNodeId": "0001",
          "CacheSize": "5 MB",
          "CacheNodeCreateTime": "2019-11-26T01:22:52.396Z",
          "SnapshotCreateTime": "2019-12-05T06:38:23Z"
        }
      ]
    },
    {
      "SnapshotName": "myreplica-backup",
      "CacheClusterId": "myreplica",
      "SnapshotStatus": "available",
      "SnapshotSource": "manual",
      "CacheNodeType": "cache.r5.large",
      "Engine": "redis",
      "EngineVersion": "5.0.5",
```

```
    "NumCacheNodes": 1,
    "PreferredAvailabilityZone": "us-west-2a",
    "CacheClusterCreateTime": "2019-11-26T00:14:52.439Z",
    "PreferredMaintenanceWindow": "sat:10:00-sat:11:00",
    "TopicArn": "arn:aws:sns:us-west-2: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": "2019-11-26T00:14:52.439Z",
        "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"
    }
  ]
}

```

Para obter mais informações, consulte [Backup e restauração ElastiCache para Redis](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [DescribeSnapshots](#) na Referência de AWS CLI Comandos.

describe-update-actions

O código de exemplo a seguir mostra como usar `describe-update-actions`.

AWS CLI

Para descrever as ações de atualização

O `describe-update-actions` exemplo a seguir retorna detalhes das ações de atualização.

```
aws elasticache describe-update-actions
```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Atualizações de autoatendimento na Amazon ElastiCache no Guia](#) do usuário do Elasticache.

- Para API obter detalhes, consulte [DescribeUpdateActions](#) na Referência de AWS CLI Comandos.

describe-user-groups

O código de exemplo a seguir mostra como usar `describe-user-groups`.

AWS CLI

Para descrever grupos de usuários

O `describe-user-groups` exemplo a seguir retorna uma lista de grupos de usuários.

```
aws elasticache describe-user-groups
```

Saída:

```

{
  "UserGroups": [
    {
      "UserGroupId": "myusergroup",
      "Status": "active",
      "Engine": "redis",
      "UserIds": [
        "default"
      ],
      "ReplicationGroups": [],
      "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
    }
  ]
}

```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [DescribeUserGroups](#) na Referência de AWS CLI Comandos.

describe-users

O código de exemplo a seguir mostra como usar describe-users.

AWS CLI

Para descrever usuários

O describe-users exemplo a seguir retorna uma lista de usuários.

```
aws elasticache describe-users
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [DescribeUsers](#) na Referência de AWS CLI Comandos.

disassociate-global-replication-group

O código de exemplo a seguir mostra como usar `disassociate-global-replication-group`.

AWS CLI

Para dissociar um cluster secundário de um grupo de replicação global

O `disassociate-global-replication-group` exemplo a seguir remove um cluster secundário de um armazenamento de dados global

```

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
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global no Guia do usuário](#) do ElastiCache.

- Para API obter detalhes, consulte [DisassociateGlobalReplicationGroup](#) na Referência de AWS CLI Comandos.

increase-node-groups-in-global-replication-group

O código de exemplo a seguir mostra como usar `increase-node-groups-in-global-replication-group`.

AWS CLI

Para aumentar o número de grupos de nós em um grupo de replicação global

O seguinte `increase-node-groups-in-global-replication-group` aumenta a contagem de grupos de nós usando o mecanismo Redis.

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --global-replication-group-id sgaui-pat-test-4 \  
  --node-group-count 6 \  
  --apply-immediately
```

Saída:

```
{  
  "GlobalReplicationGroup": {  
    "GlobalReplicationGroupId": "sgaui-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": "sgaui-test-4-0001",
        "Slots": "0-234,2420-5461"
      },
      {
        "GlobalNodeGroupId": "sgaui-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
  }
}
```

Para obter mais informações, consulte [Replicação entre AWS regiões usando o armazenamento de dados global no Guia do usuário](#) do ElastiCache.

- Para API obter detalhes, consulte [IncreaseNodeGroupsInGlobalReplicationGroup](#) na Referência de AWS CLI Comandos.

increase-replica-count

O código de exemplo a seguir mostra como usar `increase-replica-count`.

AWS CLI

Para aumentar a contagem de réplicas

O `increase-replica-count` exemplo a seguir faz uma das duas coisas. Ele pode aumentar dinamicamente o número de réplicas em um grupo de replicação do Redis (modo de cluster desativado). Ou pode aumentar dinamicamente o número de nós de réplica em um ou mais grupos de nós (fragmentos) de um grupo de replicação Redis (modo de cluster ativado). Essa operação é executada sem tempo de inatividade do cluster.

```
aws elasticache increase-replica-count \  
  --replication-group-id "my-cluster" \  
  --apply-immediately \  
  --new-replica-count 3
```

Saída:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "my-cluster",  
    "Description": " ",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "my-cluster-001",  
      "my-cluster-002",  
      "my-cluster-003",  
      "my-cluster-004"  
    ],  
    "NodeGroups": [  
      {  
        "NodeGroupId": "0001",  
        "Status": "modifying",  
        "PrimaryEndpoint": {  
          "Address": "my-  
cluster.xxxxxih.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "ReaderEndpoint": {  
          "Address": "my-cluster-  
ro.xxxxxxih.ng.0001.usw2.cache.amazonaws.com",  
          "Port": 6379  
        },  
        "NodeGroupMembers": [  
          {  
            "CacheClusterId": "my-cluster-001",
```

```

        "CacheNodeId": "0001",
        "ReadEndpoint": {
            "Address": "my-
cluster-001.xxxxxih.0001.usw2.cache.amazonaws.com",
            "Port": 6379
        },
        "PreferredAvailabilityZone": "us-west-2a",
        "CurrentRole": "primary"
    },
    {
        "CacheClusterId": "my-cluster-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
}
}

```

Para obter mais informações, consulte [Aumentando o número de réplicas em um fragmento](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [IncreaseReplicaCount](#) na Referência de AWS CLI Comandos.

list-allowed-node-type-modifications

O código de exemplo a seguir mostra como usar `list-allowed-node-type-modifications`.

AWS CLI

Para listar as modificações de nós permitidas

O `list-allowed-node-type-modifications` exemplo a seguir lista todos os tipos de nós disponíveis para os quais você pode escalar o tipo de nó atual do cluster Redis ou do grupo de replicação.

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id "my-replication-group"
```

Saída:

```
{  
  "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"  
  ]  
}
```

```
}
```

Para obter mais informações, consulte [Dimensionamento ElastiCache para clusters Redis](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [ListAllowedNodeTypeModifications](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags de um recurso.

```
aws elasticache list-tags-for-resource \  
  --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

Saída:

```
{  
  "TagList": [  
    {  
      "Key": "Project",  
      "Value": "querySpeedUp"  
    },  
    {  
      "Key": "Environment",  
      "Value": "PROD"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Listar tags usando o AWS CLI no Guia do usuário do ElastiCache](#).

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

modify-cache-cluster

O código de exemplo a seguir mostra como usar `modify-cache-cluster`.

AWS CLI

Para modificar clusters de cache

O `modify-cache-cluster` exemplo a seguir modifica as configurações do cluster especificado.

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id "my-cluster" \  
  --num-cache-nodes 1
```

Saída:

```
{  
  "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  
  }  
}
```



```
}
```

Para obter mais informações, consulte [Modificação de um ElastiCache cluster no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [ModifyCacheCluster](#) na Referência de AWS CLI Comandos.

modify-cache-parameter-group

O código de exemplo a seguir mostra como usar `modify-cache-parameter-group`.

AWS CLI

Para modificar um grupo de parâmetros de cache

O `modify-cache-parameter-group` exemplo a seguir modifica os parâmetros do grupo de parâmetros de cache especificado.

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

Saída:

```
{  
  "CacheParameterGroupName": "mygroup"  
}
```

Para obter mais informações, consulte [Modificação de um grupo de parâmetros no Guia](#) do usuário do ElastiCache.

- Para API obter detalhes, consulte [ModifyCacheParameterGroup](#) na Referência de AWS CLI Comandos.

modify-cache-subnet-group

O código de exemplo a seguir mostra como usar `modify-cache-subnet-group`.

AWS CLI

Para modificar um grupo de sub-redes de cache

O `modify-cache-subnet-group` exemplo a seguir modifica o grupo de sub-rede de cache especificado.

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name kxkxk \  
  --cache-subnet-group-description "mygroup"
```

Saída:

```
{  
  "CacheSubnetGroup": {  
    "CacheSubnetGroupName": "kxkxk",  
    "CacheSubnetGroupDescription": "mygroup",  
    "VpcId": "vpc-xxxxcdb",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-xxxxbff",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Modificar um grupo de sub-redes](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [ModifyCacheSubnetGroup](#) na Referência de AWS CLI Comandos.

modify-global-replication-group

O código de exemplo a seguir mostra como usar `modify-global-replication-group`.

AWS CLI

Para modificar um grupo de replicação global

O seguinte `modify-global-replication-group` modifica as propriedades de um grupo de replicação global, neste caso desativando o failover automático, usando o mecanismo Redis.


```
--replication-group-id mycluster \  
--node-group-count 3 \  
--apply-immediately \  
--node-groups-to-remove 0002
```

Saída

```
{  
  "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
  }
}

```

Para obter mais informações, consulte [Dimensionamento ElastiCache para clusters Redis](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [ModifyReplicationGroupShardConfiguration](#) na Referência de AWS CLI Comandos.

modify-replication-group

O código de exemplo a seguir mostra como usar `modify-replication-group`.

AWS CLI

Para modificar um grupo de replicação

O seguinte `modify-replication-group` desativa o Multi-AZ usando o mecanismo Redis.

```

aws elasticache modify-replication-group \
  --replication-group-id test-cluster \
  --no-multi-az-enabled \
  --apply-immediately

```

Saída

```

{
  "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
}
}

```

Para obter mais informações, consulte [Modificar um grupo de replicação](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [ModifyReplicationGroup](#) na Referência de AWS CLI Comandos.

modify-user-group

O código de exemplo a seguir mostra como usar `modify-user-group`.

AWS CLI

Para modificar um grupo de usuários

O `modify-user-group` exemplo a seguir adiciona um usuário ao grupo de usuários.

```

aws elasticache modify-user-group \
  --user-group-id myusergroup \
  --user-ids-to-add user1

```

Saída:

```
{
  "UserGroupId": "myusergroup",
  "Status": "modifying",
  "Engine": "redis",
  "UserIds": [
    "default"
  ],
  "PendingChanges": {
    "UserIdsToAdd": [
      "user1"
    ]
  },
  "ReplicationGroups": [],
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [ModifyUserGroup](#) na Referência de AWS CLI Comandos.

modify-user

O código de exemplo a seguir mostra como usar `modify-user`.

AWS CLI

Para modificar um usuário

O `modify-user` exemplo a seguir modifica a string de acesso de um usuário.

```
aws elasticache modify-user \
  --user-id user2 \
  --append-access-string "on ~* +@all"
```

Saída:

```
{
  "UserId": "user2",
  "UserName": "myUser",
  "Status": "modifying",
```

```

    "Engine": "redis",
    "AccessString": "on ~* +@all",
    "UserGroupIds": [],
    "Authentication": {
      "Type": "password",
      "PasswordCount": 1
    },
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:user:user2"
  }
}

```

Para obter mais informações, consulte [Autenticação de usuários com controle de acesso baseado em função \(RBAC\)](#) no Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [ModifyUser](#) na Referência de AWS CLI Comandos.

purchase-reserved-cache-nodes-offering

O código de exemplo a seguir mostra como usar `purchase-reserved-cache-nodes-offering`.

AWS CLI

Para comprar um `reserved-cache-node-offering`

O `purchase-reserved-cache-nodes-offering` exemplo a seguir permite que você compre uma oferta de nó de cache reservado.

```

aws elasticache purchase-reserved-cache-nodes-offering \
  --reserved-cache-nodes-offering-id xxxxxxxx-4da5-4b90-b92d-929fbd7abed2

```

Saída

```

{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Obter informações sobre as ofertas de nós reservados](#) no Guia do usuário do ElastiCache Redis ou [Obter informações sobre as ofertas de nós reservados](#) no Guia do usuário do ElastiCache Memcached.

- Para API obter detalhes, consulte [PurchaseReservedCacheNodesOffering](#) na Referência de AWS CLI Comandos.

reboot-cache-cluster

O código de exemplo a seguir mostra como usar `reboot-cache-cluster`.

AWS CLI

Para reinicializar um cluster de cache

O `reboot-cache-cluster` exemplo a seguir reinicializa alguns ou todos os nós de cache em um cluster provisionado. Essa operação aplica qualquer grupo de parâmetros de cache modificado ao cluster. A operação de reinicialização ocorre o mais rápido possível e resulta em uma interrupção momentânea do cluster. Durante a reinicialização, o status do cluster é definido como `REBOOTING`.

```

aws elasticache reboot-cache-cluster \
  --cache-cluster-id "my-cluster-001" \
  --cache-node-ids-to-reboot "0001"

```

Saída:

```

{
  "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
}
}

```

Para obter mais informações, consulte Reinicializando um cluster < <https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/clusters.rebooting.html> no Guia do usuário do ElastiCache.

- Para API obter detalhes, consulte [RebootCacheCluster](#) na Referência de AWS CLI Comandos.

reset-cache-parameter-group

O código de exemplo a seguir mostra como usar `reset-cache-parameter-group`.

AWS CLI

Para redefinir um grupo de parâmetros de cache

O `reset-cache-parameter-group` exemplo a seguir modifica os parâmetros de um grupo de parâmetros de cache para o valor padrão do mecanismo ou do sistema. Você pode redefinir parâmetros específicos enviando uma lista de nomes de parâmetros. Para redefinir todo o grupo de parâmetros do cache, especifique os `--cache-parameter-group-name` parâmetros `--reset-all-parameters` e.

```
aws elasticache reset-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --reset-all-parameters
```

Saída:

```
{  
  "CacheParameterGroupName": "mygroup"  
}
```

- Para API obter detalhes, consulte [ResetCacheParameterGroup](#) na Referência de AWS CLI Comandos.

start-migration

O código de exemplo a seguir mostra como usar `start-migration`.

AWS CLI

Para iniciar uma migração

O seguinte `start-migration` migra seus dados do Redis auto-hospedado na Amazon para a EC2 Amazon ElastiCache, usando o mecanismo do Redis.

```
aws elasticache start-migration \  
  --replication-group-id test \  
  --
```

--customer-node-endpoint-

```
list "Address='test.g2xbih.ng.0001.usw2.cache.amazonaws.com',Port=6379"
```

Saída

```
{
  "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
}
}

```

Para obter mais informações, consulte [Migração on-line para](#) o ElastiCache Guia do usuário do Elasticache.

- Para API obter detalhes, consulte [StartMigration](#) na Referência de AWS CLI Comandos.

test-failover

O código de exemplo a seguir mostra como usar `test-failover`.

AWS CLI

Para testar o failover de um grupo de nós

O `test-failover` exemplo a seguir testa o failover automático no grupo de nós especificado (chamado de fragmento no console) em um grupo de replicação (chamado de cluster no console).

```
aws elasticache test-failover /  
  --replication-group-id "mycluster" /  
  --node-group-id "0001"
```

Saída:

```
{  
  "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
}
}

```

- Para API obter detalhes, consulte [TestFailover](#) na Referência de AWS CLI Comandos.

MediaStore exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaStore.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-container

O código de exemplo a seguir mostra como usar `create-container`.

AWS CLI

Para criar um contêiner

O `create-container` exemplo a seguir cria um novo contêiner vazio.

```
aws mediastore create-container --container-name ExampleContainer
```

Saída:

```
{
  "Container": {
    "AccessLoggingEnabled": false,
    "CreationTime": 1563557265,
    "Name": "ExampleContainer",
    "Status": "CREATING",
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
  }
}
```

Para obter mais informações, consulte [Criação de um contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateContainer](#) na Referência de AWS CLI Comandos.

delete-container-policy

O código de exemplo a seguir mostra como usar `delete-container-policy`.

AWS CLI

Para excluir uma política de contêiner

O `delete-container-policy` exemplo a seguir exclui a política atribuída ao contêiner especificado. Quando a política é excluída, o AWS Elemental atribui MediaStore automaticamente a política padrão ao contêiner.

```
aws mediastore delete-container-policy \  
  --container-name LiveEvents
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteContainerPolicy](#) na MediaStore API referência AWS Elemental.

- Para API obter detalhes, consulte [DeleteContainerPolicy](#) na Referência de AWS CLI Comandos.

delete-container

O código de exemplo a seguir mostra como usar `delete-container`.

AWS CLI

Para excluir um contêiner

O `delete-container` exemplo a seguir exclui o contêiner especificado. Você só pode excluir um contêiner se ele não tiver objetos.

```
aws mediastore delete-container \  
  --container-name=ExampleLiveDemo
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um contêiner no Guia](#) do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeleteContainer](#) na Referência de AWS CLI Comandos.

delete-cors-policy

O código de exemplo a seguir mostra como usar `delete-cors-policy`.

AWS CLI

Para excluir uma CORS política

O `delete-cors-policy` exemplo a seguir exclui a política de compartilhamento de recursos de origem cruzada (CORS) atribuída ao contêiner especificado.

```
aws mediastore delete-cors-policy \  
  --container-name ExampleContainer
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo uma CORS política no Guia](#) do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeleteCorsPolicy](#) na Referência de AWS CLI Comandos.

delete-lifecycle-policy

O código de exemplo a seguir mostra como usar `delete-lifecycle-policy`.

AWS CLI

Para excluir uma política de ciclo de vida de objetos

O `delete-lifecycle-policy` exemplo a seguir exclui a política de ciclo de vida do objeto anexada ao contêiner especificado. Essa alteração pode levar até 20 minutos para entrar em vigor.

```
aws mediastore delete-lifecycle-policy \  
  --container-name LiveEvents
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo uma política de ciclo de vida de objetos](#) no Guia do usuário do AWS Elemental MediaStore .

- Para API obter detalhes, consulte [DeleteLifecyclePolicy](#) na Referência de AWS CLI Comandos.

describe-container

O código de exemplo a seguir mostra como usar `describe-container`.

AWS CLI

Para ver os detalhes de um contêiner

O `describe-container` exemplo a seguir exibe os detalhes do contêiner especificado.

```
aws mediastore describe-container \  
  --container-name ExampleContainer
```

Saída:

```
{  
  "Container": {  
    "CreationTime": 1563558086,  
    "AccessLoggingEnabled": false,  
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/  
ExampleContainer",  
    "Status": "ACTIVE",  
    "Name": "ExampleContainer",  
    "Endpoint": "https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com"  
  }  
}
```

Para obter mais informações, consulte [Visualizando os detalhes de um contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribeContainer](#) na Referência de AWS CLI Comandos.

describe-object

O código de exemplo a seguir mostra como usar `describe-object`.

AWS CLI

Para visualizar uma lista de objetos e pastas em um contêiner específico

O `describe-object` exemplo a seguir exibe itens (objetos e pastas) armazenados em um contêiner específico.

```
aws mediastore-data describe-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path /folder_name/file1234.jpg
```

Saída:

```
{  
  "ContentType": "image/jpeg",
```

```

    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3"
  }

```

Para obter mais informações, consulte [Visualizando os detalhes de um objeto](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribeObject](#) na Referência de AWS CLI Comandos.

get-container-policy

O código de exemplo a seguir mostra como usar `get-container-policy`.

AWS CLI

Para ver uma política de contêiner

O `get-container-policy` exemplo a seguir exibe a política baseada em recursos do contêiner especificado.

```

aws mediastore get-container-policy \
  --container-name ExampleLiveDemo

```

Saída:

```

{
  "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"
            }
        }
    ]
}

```

Para obter mais informações, consulte [Visualizando uma política de contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [GetContainerPolicy](#) na Referência de AWS CLI Comandos.

get-cors-policy

O código de exemplo a seguir mostra como usar `get-cors-policy`.

AWS CLI

Para ver uma CORS política

O `get-cors-policy` exemplo a seguir exibe a política de compartilhamento de recursos de origem cruzada (CORS) atribuída ao contêiner especificado.

```

aws mediastore get-cors-policy \
  --container-name ExampleContainer \
  --region us-west-2

```

Saída:

```

{
  "CorsPolicy": [
    {
      "AllowedMethods": [
        "GET",
        "HEAD"
      ],
      "MaxAgeSeconds": 3000,
      "AllowedOrigins": [
        ""
      ]
    }
  ]
}

```

```
    ],
    "AllowedHeaders": [
        ""
    ]
}
]
```

Para obter mais informações, consulte [Visualizando uma CORS política](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [GetCorsPolicy](#) na Referência de AWS CLI Comandos.

get-lifecycle-policy

O código de exemplo a seguir mostra como usar `get-lifecycle-policy`.

AWS CLI

Para visualizar uma política de ciclo de vida de objetos

O `get-lifecycle-policy` exemplo a seguir exibe a política de ciclo de vida do objeto anexada ao contêiner especificado.

```
aws mediastore get-lifecycle-policy \
  --container-name LiveEvents
```

Saída:

```
{
  "LifecyclePolicy": {
    "rules": [
      {
        "definition": {
          "path": [
            {
              "prefix": "Football/"
            },
            {
              "prefix": "Baseball/"
            }
          ]
        }
      ]
    }
  }
}
```



```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Visualizando uma lista de contêineres](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListContainers](#) na Referência de AWS CLI Comandos.

list-items

O código de exemplo a seguir mostra como usar `list-items`.

AWS CLI

Exemplo 1: Para visualizar uma lista de objetos e pastas em um contêiner específico

O `list-items` exemplo a seguir exibe itens (objetos e pastas) armazenados no contêiner especificado.

```
aws mediastore-data list-items \
```

```
--endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

Saída:

```
{
  "Items": [
    {
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379,
      "Name": "filename.jpg",
      "Type": "OBJECT",
      "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
      "ContentLength": 3784
    },
    {
      "Type": "FOLDER",
      "Name": "ExampleLiveDemo"
    }
  ]
}
```

Exemplo 2: Para visualizar uma lista de objetos e pastas em uma pasta específica

O `list-items` exemplo a seguir exibe itens (objetos e pastas) armazenados em uma pasta específica.

```
aws mediastore-data list-items \  
--endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

Saída:

```
{
  "Items": [
    {
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379,
      "Name": "filename.jpg",
      "Type": "OBJECT",
      "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
      "ContentLength": 3784
    }
  ]
}
```

```
    },
    {
      "Type": "FOLDER",
      "Name": "ExampleLiveDemo"
    }
  ]
}
```

Para obter mais informações, consulte [Visualizando uma lista de objetos](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListItems](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um contêiner

O `list-tags-for-resource` exemplo a seguir exibe as chaves e os valores de tag atribuídos ao contêiner especificado.

```
aws mediastore list-tags-for-resource \
  --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

Saída:

```
{
  "Tags": [
    {
      "Value": "Test",
      "Key": "Environment"
    },
    {
      "Value": "West",
      "Key": "Region"
    }
  ]
}
```

Para obter mais informações, consulte [ListTagsForResource](#) na MediaStore API Referência AWS Elemental.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-container-policy

O código de exemplo a seguir mostra como usar `put-container-policy`.

AWS CLI

Para editar uma política de contêiner

O `put-container-policy` exemplo a seguir atribui uma política diferente ao contêiner especificado. Neste exemplo, a política atualizada é definida em um arquivo chamado `LiveEventsContainerPolicy.json`.

```
aws mediastore put-container-policy \  
  --container-name LiveEvents \  
  --policy file:///LiveEventsContainerPolicy.json
```

Este comando não produz saída.

Para obter mais informações, consulte [Editando uma política de contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [PutContainerPolicy](#) na Referência de AWS CLI Comandos.

put-cors-policy

O código de exemplo a seguir mostra como usar `put-cors-policy`.

AWS CLI

Exemplo 1: Para adicionar uma CORS política

O `put-cors-policy` exemplo a seguir adiciona uma política de compartilhamento de recursos (CORS) de origem cruzada ao contêiner especificado. O conteúdo da CORS política está no arquivo chamado `corsPolicy.json`.

```
aws mediastore put-cors-policy \  
  --container-name ExampleContainer \  
  --policy file:///corsPolicy.json
```



```
--cors-policy file://corsPolicy.json
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma CORS política a um contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

Exemplo 2: Para editar uma CORS política

O `put-cors-policy` exemplo a seguir atualiza a política de compartilhamento de recursos de origem cruzada (CORS) atribuída ao contêiner especificado. O conteúdo da CORS política atualizada está no arquivo chamado `corsPolicy2.json`.

Para obter mais informações, consulte [Editando uma CORS política](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [PutCorsPolicy](#) na Referência de AWS CLI Comandos.

put-lifecycle-policy

O código de exemplo a seguir mostra como usar `put-lifecycle-policy`.

AWS CLI

Para criar uma política de ciclo de vida de objetos

O `put-lifecycle-policy` exemplo a seguir anexa uma política de ciclo de vida do objeto ao contêiner especificado. Isso permite que você especifique por quanto tempo o serviço deve armazenar objetos em seu contêiner. MediaStore exclui objetos no contêiner quando eles atingem sua data de expiração, conforme indicado na política, que está no arquivo nomeado `LiveEventsLifecyclePolicy.json`.

```
aws mediastore put-lifecycle-policy \  
  --container-name ExampleContainer \  
  --lifecycle-policy file://ExampleLifecyclePolicy.json
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma política de ciclo de vida de objetos a um contêiner](#) no Guia do usuário do AWS Elemental MediaStore .

- Para API obter detalhes, consulte [PutLifecyclePolicy](#) na Referência de AWS CLI Comandos.

put-object

O código de exemplo a seguir mostra como usar `put-object`.

AWS CLI

Para fazer o upload de um objeto

O `put-object` exemplo a seguir carrega um objeto no contêiner especificado. Você pode especificar um caminho de pasta em que o objeto será salvo no contêiner. Se a pasta já existir, o AWS Elemental MediaStore armazena o objeto na pasta. Se a pasta não existir, o serviço a cria e, em seguida, armazena o objeto na pasta.

```
aws mediastore-data put-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --body README.md \  
  --path /folder_name/README.md \  
  --cache-control "max-age=6, public" \  
  --content-type binary/octet-stream
```

Saída:

```
{  
  "ContentSHA256":  
    "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",  
  "StorageClass": "TEMPORAL",  
  "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"  
}
```

Para obter mais informações, consulte [Carregando um objeto no Guia](#) do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [PutObject](#) na Referência de AWS CLI Comandos.

start-access-logging

O código de exemplo a seguir mostra como usar `start-access-logging`.

AWS CLI

Para habilitar o registro de acesso em um contêiner

O `start-access-logging` exemplo a seguir habilita o registro de acesso no contêiner especificado.

```
aws mediastore start-access-logging \  
  --container-name LiveEvents
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando o registro de acesso para um contêiner](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [StartAccessLogging](#) na Referência de AWS CLI Comandos.

stop-access-logging

O código de exemplo a seguir mostra como usar `stop-access-logging`.

AWS CLI

Para desativar o registro de acesso em um contêiner

O `stop-access-logging` exemplo a seguir desativa o registro de acesso no contêiner especificado.

```
aws mediastore stop-access-logging \  
  --container-name LiveEvents
```

Este comando não produz saída.

Para obter mais informações, consulte Como [desativar o registro de acesso para um contêiner no Guia](#) do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [StopAccessLogging](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um contêiner

O `tag-resource` exemplo a seguir adiciona chaves e valores de tag ao contêiner especificado.

```
aws mediastore tag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value": "Test"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [TagResource](#) na MediaStore API Referência AWS Elemental.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover etiquetas de um contêiner

O `untag-resource` exemplo a seguir remove a chave de tag especificada e seu valor associado de um contêiner.

```
aws mediastore untag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tag-keys Region
```

Este comando não produz saída.

Para obter mais informações, consulte [UntagResource](#) na MediaStore API Referência AWS Elemental.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

EMR Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonEMR.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-instance-fleet

O código de exemplo a seguir mostra como usar `add-instance-fleet`.

AWS CLI

Para adicionar uma frota de instâncias de tarefas a um cluster

Este exemplo adiciona uma nova frota de instâncias de tarefas ao cluster especificado.

Comando:

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet  
InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification='{Timeo
```

Saída:

```
{  
  "ClusterId": "j-12ABCDEFGHI34JK",  
  "InstanceFleetId": "if-23ABCDEFGHI45JJ"  
}
```

- Para API obter detalhes, consulte [AddInstanceFleet](#) na Referência de AWS CLI Comandos.

add-steps

O código de exemplo a seguir mostra como usar `add-steps`.

AWS CLI

1. Para adicionar JAR etapas personalizadas a um 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
```

Parâmetros necessários:

Jar

Parâmetros opcionais:

Type, Name, ActionOnFailure, Args

Saída:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

2. Para adicionar etapas de streaming a um 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]
```

Parâmetros necessários:

```
Type, Args
```

Parâmetros opcionais:

```
Name, ActionOnFailure
```

JSONequivalente (conteúdo de 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 os argumentos devem incluir opções e valores como seus próprios itens na lista.

Comando (usando step.json):

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file:///./step.json
```

Saída:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

3. Para adicionar uma etapa de streaming com vários arquivos a um cluster (JSONsamente)

JSON(vários arquivos.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
```

Parâmetros necessários:

Type, Args

Parâmetros opcionais:

Name, ActionOnFailure

Saída:

```
{
  "StepIds":[
    "s-XXXXXXXX",
  ]
}
```

4. Para adicionar etapas do Hive a um 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]
```

Parâmetros necessários:

Type, Args

Parâmetros opcionais:

Name, ActionOnFailure

Saída:

```
{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

5. Para adicionar etapas do Pig a um 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]
```

Parâmetros necessários:

Type, Args

Parâmetros opcionais:

Name, ActionOnFailure

Saída:

```
{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

6. Para adicionar etapas do Impala a um 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
```

Parâmetros necessários:

Type, Args

Parâmetros opcionais:

Name, ActionOnFailure

Saída:

```
{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

- Para API obter detalhes, consulte [AddSteps](#) na Referência de AWS CLI Comandos.

add-tags

O código de exemplo a seguir mostra como usar `add-tags`.

AWS CLI

1. Para adicionar tags a um cluster

Comando:

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
address="123 East NW Seattle"
```

Saída:

```
None
```

2. Para listar as tags de um cluster

--Comando:

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

Saída:

```
[
  {
    "Value": "male",
    "Key": "sex"
  },
  {
    "Value": "123 East NW Seattle",
    "Key": "address"
  },
  {
    "Value": "John Doe",
    "Key": "name"
  },
  {
    "Value": "29",
    "Key": "age"
  }
]
```

```
}
]
```

- Para API obter detalhes, consulte [AddTags](#) na Referência de AWS CLI Comandos.

create-cluster-examples

O código de exemplo a seguir mostra como usar `create-cluster-examples`.

AWS CLI

A maioria dos exemplos a seguir pressupõe que você especificou sua função de EMR serviço da Amazon e seu perfil de EC2 instância da Amazon. Se você não tiver feito isso, deverá especificar cada IAM função necessária ou usar o `--use-default-roles` parâmetro ao criar seu cluster. Para obter mais informações sobre a especificação de IAM funções, consulte [Configurar IAM funções para Amazon EMR Permissions to AWS Services](#) no Amazon EMR Management Guide.

Exemplo 1: Para criar um cluster

O `create-cluster` exemplo a seguir cria um EMR cluster simples.

```
aws emr create-cluster \
  --release-label emr-5.14.0 \
  --instance-type m4.large \
  --instance-count 2
```

Este comando não produz saída.

Exemplo 2: Para criar um EMR cluster da Amazon com InstanceProfile funções ServiceRole e padrões

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa a `--instance-groups` configuração.

```
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
```

Exemplo 3: Para criar um EMR cluster da Amazon que usa uma frota de instâncias

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa a `--instance-fleets` configuração, especificando dois tipos de instância para cada frota e duas EC2 sub-redes.

```
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
```

Exemplo 4: Para criar um cluster com funções padrão

O `create-cluster` exemplo a seguir usa o `--use-default-roles` parâmetro para especificar a função de serviço padrão e o perfil da instância.

```
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
```

Exemplo 5: Para criar um cluster e especificar os aplicativos a serem instalados

O `create-cluster` exemplo a seguir usa o `--applications` parâmetro para especificar os aplicativos que a Amazon EMR instala. Este exemplo instala o Hadoop, o Hive e o 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
```

Exemplo 6: Para criar um cluster que inclua o Spark

O exemplo a seguir instala o 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
```

Exemplo 7: Para especificar um personalizado AMI a ser usado para instâncias de cluster

O create-cluster exemplo a seguir cria uma instância de cluster baseada no Amazon Linux AMI com 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 \
  --instance-count 2 \
  --instance-type m4.Large
```

Exemplo 8: Para personalizar as configurações do aplicativo

Os exemplos a seguir usam o --configurations parâmetro para especificar um arquivo de JSON configuração que contém personalizações de aplicativos para o Hadoop. Para obter mais informações, consulte [Configurando aplicativos](#) no Amazon EMR Release Guide.

Conteúdo de configurations.json:

```
[
  {
    "Classification": "mapred-site",
    "Properties": {
      "mapred.tasktracker.map.tasks.maximum": 2
    }
  }
]
```

```

    },
    {
      "Classification": "hadoop-env",
      "Properties": {},
      "Configurations": [
        {
          "Classification": "export",
          "Properties": {
            "HADOOP_DATANODE_HEAPSIZE": 2048,
            "HADOOP_NAMENODE_OPTS": "-XX:GCTimeRatio=19"
          }
        }
      ]
    }
  ]
}
]

```

O exemplo a seguir faz referência `configurations.json` a um arquivo local.

```

aws emr create-cluster \
  --configurations file://configurations.json \
  --release-label emr-5.9.0 \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
  --auto-terminate

```

O exemplo a seguir faz referência `configurations.json` a um arquivo no 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

```

Exemplo 9: Para criar um cluster com grupos de instâncias mestre, principal e de tarefas

O `create-cluster` exemplo a seguir é usado `--instance-groups` para especificar o tipo e o número de EC2 instâncias a serem usadas para grupos de instâncias mestre, principal e de tarefas.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --instance-
groups Name=Master, InstanceGroupType=MASTER, InstanceType=m4.Large, InstanceCount=1 Name=Core,
```

Exemplo 10: Para especificar que um cluster deve ser encerrado após a conclusão de todas as etapas

O `create-cluster` exemplo a seguir é usado `--auto-terminate` para especificar que o cluster deve ser desligado automaticamente após a conclusão de todas as etapas.

```
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
```

Exemplo 11: Para especificar detalhes da configuração do cluster, como o par de EC2 chaves da Amazon, a configuração da rede e os grupos de segurança

O `create-cluster` exemplo a seguir cria um cluster com o par de EC2 chaves da Amazon chamado `myKey` e um perfil de instância personalizado chamado `myProfile`. Os pares de chaves são usados para autorizar SSH conexões com os nós do cluster, geralmente o nó principal. Para obter mais informações, consulte [Usar um par de EC2 chaves da Amazon para SSH credenciais](#) no Guia de EMR gerenciamento da Amazon.

```
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
```

O exemplo a seguir cria um cluster em uma VPC sub-rede da 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

```

O exemplo a seguir cria um cluster na zona de us-east-1b disponibilidade.

```

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

```

O exemplo a seguir cria um cluster e especifica somente os grupos de segurança EMR gerenciados pela 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

```

O exemplo a seguir cria um cluster e especifica somente grupos de EC2 segurança adicionais da Amazon.

```

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

```

O exemplo a seguir cria um cluster e especifica os grupos de segurança EMR -Managed, bem como os grupos de segurança adicionais.

```

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

```

O exemplo a seguir cria um cluster em uma sub-rede VPC privada e usa um grupo de EC2 segurança específico da Amazon para permitir o acesso aos EMR serviços da Amazon, o que é necessário para clusters em sub-redes privadas.

```

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

```

O exemplo a seguir especifica os parâmetros de configuração do grupo de segurança usando um JSON arquivo chamado `ec2_attributes.json` que é armazenado localmente. NOTE: JSON os argumentos devem incluir opções e valores como seus próprios itens na lista.

```

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

```

Conteúdo de `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"]
  }
]

```

Exemplo 12: Para habilitar a depuração e especificar um log URI

O `create-cluster` exemplo a seguir usa o `--enable-debugging` parâmetro, que permite que você visualize arquivos de log com mais facilidade usando a ferramenta de depuração no console da Amazon. EMR O `--log-uri` parâmetro é obrigatório com `--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

```

Exemplo 13: Para adicionar tags ao criar um cluster

As tags são pares de valores-chave que ajudam você a identificar e gerenciar clusters. O `create-cluster` exemplo a seguir usa o `--tags` parâmetro para criar três tags para um cluster, uma com o nome da chave `name` e o valor `Shirley Rodriguez`, uma segunda com o nome da chave `age` e o valor `29` e uma terceira tag com o nome da chave `department` e o valor `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

```

O exemplo a seguir lista as tags aplicadas a um cluster.

```
aws emr describe-cluster \
  --cluster-id j-XXXXXXYY \
  --query Cluster.Tags
```

Exemplo 14: Para usar uma configuração de segurança que habilite a criptografia e outros recursos de segurança

O `create-cluster` exemplo a seguir usa o `--security-configuration` parâmetro para especificar uma configuração de segurança para um EMR cluster. Você pode usar configurações de segurança com a Amazon EMR versão 4.8.0 ou posterior.

```
aws emr create-cluster \
  --instance-type m4.large \
  --release-label emr-5.9.0 \
  --security-configuration mySecurityConfiguration
```

Exemplo 15: Para criar um cluster com volumes de EBS armazenamento adicionais configurados para os grupos de instâncias

Ao especificar EBS volumes adicionais, os seguintes argumentos são necessários: `VolumeType`, `SizeInGB` se `EbsBlockDeviceConfigs` for especificado.

O `create-cluster` exemplo a seguir cria um cluster com vários EBS volumes anexados às EC2 instâncias no grupo de instâncias principal.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --use-default-roles \
  --instance-
groups InstanceGroupType=MASTER, InstanceCount=1, InstanceType=d2.xlarge
'InstanceGroupType=CORE, InstanceCount=2, InstanceType=d2.xlarge, EbsConfiguration={EbsOptimiz
{VolumeSpecification={VolumeType=io1, SizeInGB=100, Iops=100}, VolumesPerInstance=4}}'
\
  --auto-terminate
```

O exemplo a seguir cria um cluster com vários EBS volumes anexados às EC2 instâncias no grupo de instâncias mestre.

```
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

```

Exemplo 16: Para criar um cluster com uma política de escalabilidade automática

Você pode anexar políticas de escalabilidade automática aos grupos de instâncias principais e de tarefas usando a Amazon EMR versão 4.0 e posterior. A política de escalabilidade automática adiciona e remove EC2 instâncias dinamicamente em resposta a uma métrica da Amazon CloudWatch. Para obter mais informações, consulte Usando a escalabilidade automática na Amazon EMR < <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html> > no Guia de gerenciamento da Amazon EMR.

Ao anexar uma política de escalabilidade automática, você também deve especificar a função padrão para uso da escalabilidade automática. `--auto-scaling-role EMR_AutoScaling_DefaultRole`

O `create-cluster` exemplo a seguir especifica a política de escalabilidade automática para o grupo de CORE instâncias usando o `AutoScalingPolicy` argumento com uma JSON estrutura incorporada, que especifica a configuração da política de escalabilidade. Grupos de instâncias com uma JSON estrutura incorporada devem ter toda a coleção de argumentos entre aspas simples. Usar aspas simples é opcional para grupos de instâncias sem uma JSON estrutura incorporada.

```

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

```

O exemplo a seguir usa um JSON arquivo, `instancegroupconfig.json`, para especificar a configuração de todos os grupos de instâncias em um cluster. O JSON arquivo especifica a configuração automática da política de escalabilidade para o grupo de instâncias principal.

```
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
```

Conteúdo de `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}"
      }
    ]
  }
}
]

```

Exemplo 17: Adicionar JAR etapas personalizadas ao criar um cluster

O `create-cluster` exemplo a seguir adiciona etapas especificando um JAR arquivo armazenado no Amazon S3. As etapas enviam o trabalho para um cluster. A função principal definida no JAR arquivo é executada depois que as EC2 instâncias são provisionadas, todas as ações de bootstrap são executadas e os aplicativos são instalados. As etapas são especificadas usando `Type=CUSTOM_JAR`.

As JAR etapas personalizadas exigem o `Jar=` parâmetro, que especifica o caminho e o nome do arquivo do JAR. Os parâmetros opcionais são `Type`, `NameActionOnFailure`, `Args`, `MainClass` e. Se a classe principal não for especificada, o JAR arquivo deverá especificar `Main-Class` em seu arquivo de manifesto.

```

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

```

Exemplo 18: Para adicionar etapas de streaming ao criar um cluster

Os `create-cluster` exemplos a seguir adicionam uma etapa de streaming a um cluster que termina após a execução de todas as etapas. As etapas de streaming exigem parâmetros `Type` e `Args`. Os parâmetros opcionais das etapas de streaming são `Name`, `ActionOnFailure` e `e`.

O exemplo a seguir especifica a etapa em linha.

```
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
```

O exemplo a seguir usa um arquivo de JSON configuração armazenado localmente chamado `multiplefiles.json`. A JSON configuração especifica vários arquivos. Para especificar vários arquivos em uma etapa, você deve usar um arquivo de JSON configuração para especificar a etapa. JSONOs argumentos devem incluir opções e valores como seus próprios itens na lista.

```
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
```

Conteúdo de `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"
}
]

```

Exemplo 19: Para adicionar etapas do Hive ao criar um cluster

O exemplo a seguir adiciona etapas do Hive ao criar um cluster. As etapas do Hive requerem parâmetros Type e. Args Os parâmetros opcionais das etapas do Hive são 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

```

Exemplo 20: Para adicionar etapas do Pig ao criar um cluster

O exemplo a seguir adiciona etapas do Pig ao criar um cluster. Os parâmetros necessários para as etapas do porco são Type e. Args Os parâmetros opcionais do Pig Steps são 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
```

Exemplo 21: Para adicionar ações de bootstrap

O `create-cluster` exemplo a seguir executa duas ações de bootstrap definidas como scripts que são armazenados no 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
```

Exemplo 22: Para permitir EMRFS uma visualização consistente e personalizar as `RetryPeriod` configurações `RetryCount` e

O `create-cluster` exemplo a seguir especifica a contagem e o período de novas tentativas para EMRFS uma visualização consistente. O argumento `Consistent=true` é obrigatório.

```
aws emr create-cluster \  
  --instance-type m4.large \  
  --release-label emr-5.9.0 \  
  --emrfs Consistent=true,RetryCount=6,RetryPeriod=30
```

O exemplo a seguir especifica a mesma EMRFS configuração do exemplo anterior, usando um arquivo de JSON configuração armazenado localmente chamado `emrfsconfig.json`.

```
aws emr create-cluster \  
  --instance-type m4.large \  
  --release-label emr-5.9.0 \  
  --emrfs file://emrfsconfig.json
```

Conteúdo de `emrfsconfig.json`:

```
{  
  "Consistent": true,
```

```

    "RetryCount": 6,
    "RetryPeriod": 30
  }

```

Exemplo 23: Para criar um cluster com o Kerberos configurado

Os `create-cluster` exemplos a seguir criam um cluster usando uma configuração de segurança com o Kerberos habilitado e estabelecem os parâmetros do Kerberos para o cluster usando `--kerberos-attributes`

O comando a seguir especifica os atributos do Kerberos para o cluster em linha.

```

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

```

O comando a seguir especifica os mesmos atributos, mas faz referência a um JSON arquivo armazenado localmente chamado `kerberos_attributes.json`. Neste exemplo, o arquivo é salvo no mesmo diretório em que você executa o comando. Você também pode referenciar um arquivo de configuração salvo no 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

```

Conteúdo de `kerberos_attributes.json`:

```

{
  "Realm": "EC2.INTERNAL",
  "KdcAdminPassword": "123",
  "CrossRealmTrustPrincipalPassword": "123",
}

```

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa a `--instance-groups` configuração e tem uma política de escalabilidade gerenciada.

```
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}'
```

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa o `--log-encryption-kms-key-id` para definir o ID da KMS chave utilizado para criptografia de 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,
```

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa a configuração `--placement-group-configs` para colocar nós principais em um cluster de alta disponibilidade (HA) dentro de um grupo de EC2 posicionamento usando a estratégia de SPREAD posicionamento.

```
aws emr create-cluster \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-
groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m4.large InstanceGroupType=CORE,
  \
  --placement-group-configs InstanceRole=MASTER
```

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa a configuração `--auto-termination-policy` para estabelecer um limite automático de encerramento de inatividade para o 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
```

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa o “`--os-release-label`” para definir uma versão do Amazon Linux para lançamento do 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
```

Exemplo 24: Para especificar os atributos de um volume EBS raiz: tamanho, iops e taxa de transferência para instâncias de cluster criadas com as EMR versões 6.15.0 e posteriores

O `create-cluster` exemplo a seguir cria um EMR cluster da Amazon que usa atributos de volume raiz para configurar especificações de volumes raiz para as EC2 instâncias.

```
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
```

- Para API obter detalhes, consulte [CreateClusterExamples](#) na Referência de AWS CLI Comandos.

create-default-roles

O código de exemplo a seguir mostra como usar `create-default-roles`.

AWS CLI

1. Para criar a IAM função padrão para EC2

Comando:

```
aws emr create-default-roles
```

Saída:

If the role already exists then the command returns nothing.

If the role does not exist then the output will be:

```
[
  {
    "RolePolicy": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": [
            "cloudwatch:*",
            "dynamodb:*",
            "ec2:Describe*",
            "elasticmapreduce:Describe*",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListInstances",
            "elasticmapreduce:ListSteps",
            "kinesis:CreateStream",
            "kinesis>DeleteStream",
            "kinesis:DescribeStream",
            "kinesis:GetRecords",
            "kinesis:GetShardIterator",
            "kinesis:MergeShards",
            "kinesis:PutRecord",
            "kinesis:SplitShard",
            "rds:Describe*",
            "s3:*",
```

```

        "sdb:*",
        "sns:*",
        "sqs:*"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
},
"Role": {
    "AssumeRolePolicyDocument": {
        "Version": "2008-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Sid": "",
                "Effect": "Allow",
                "Principal": {
                    "Service": "ec2.amazonaws.com"
                }
            }
        ]
    },
    "RoleId": "AR0AIQ5SIQUGL5KMYBJX6",
    "CreateDate": "2015-06-09T17:09:04.602Z",
    "RoleName": "EMR_EC2_DefaultRole",
    "Path": "/",
    "Arn": "arn:aws:iam::176430881729:role/EMR_EC2_DefaultRole"
}
},
{
    "RolePolicy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:AuthorizeSecurityGroupIngress",
                    "ec2:CancelSpotInstanceRequests",
                    "ec2:CreateSecurityGroup",
                    "ec2:CreateTags",
                    "ec2>DeleteTags",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeAccountAttributes",
                    "ec2:DescribeInstances",

```

```

        "ec2:DescribeInstanceStatus",
        "ec2:DescribeKeyPairs",
        "ec2:DescribePrefixLists",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcs",
        "ec2:ModifyImageAttribute",
        "ec2:ModifyInstanceAttribute",
        "ec2:RequestSpotInstances",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListInstanceProfiles",
        "iam:ListRolePolicies",
        "iam:PassRole",
        "s3:CreateBucket",
        "s3:Get*",
        "s3:List*",
        "sdb:BatchPutAttributes",
        "sdb:Select",
        "sqs:CreateQueue",
        "sqs>Delete*",
        "sqs:GetQueue*",
        "sqs:ReceiveMessage"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
},
"Role": {
    "AssumeRolePolicyDocument": {
        "Version": "2008-10-17",
        "Statement": [
            {
                "Action": "sts:AssumeRole",
                "Sid": "",

```



```

        "Effect": "Allow",
        "Principal": {
            "Service": "elasticmapreduce.amazonaws.com"
        }
    }
]
},
"RoleId": "AROAI3SRVPPVSRDLARBPY",
"CreateDate": "2015-06-09T17:09:10.401Z",
"RoleName": "EMR_DefaultRole",
"Path": "/",
"Arn": "arn:aws:iam::176430881729:role/EMR_DefaultRole"
}
}
]

```

- Para API obter detalhes, consulte [CreateDefaultRoles](#) na Referência de AWS CLI Comandos.

create-security-configuration

O código de exemplo a seguir mostra como usar `create-security-configuration`.

AWS CLI

1. Para criar uma configuração de segurança com criptografia em trânsito habilitada PEM para provedor de certificados e criptografia em repouso habilitada com SSE -S3 para criptografia S3 e AWS- KMS para provedor de chave de disco local

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"
            }
        }
    }
}'

```

Saída:

```

{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}

```

JSONequivalente (conteúdo de 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 (usando security_configuration.json):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-configuration file:///./security_configuration.json
```

Saída:

```
{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}
```

2. Para criar uma configuração de segurança com o Kerberos habilitado usando confiança dedicada ao cluster KDC e entre regiões

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"
        }
      }
    }
  }
}'
```

Saída:

```
{
  "CreationDateTime": 1490225558.982,
```

```
"Name": "MySecurityConfig"
}
```

JSONequivalente (conteúdo de 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 (usando security_configuration.json):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

Saída:

```
{
  "CreationDateTime": 1490225558.982,
  "Name": "MySecurityConfig"
}
```

- Para API obter detalhes, consulte [CreateSecurityConfiguration](#) na Referência de AWS CLI Comandos.

delete-security-configuration

O código de exemplo a seguir mostra como usar delete-security-configuration.

AWS CLI

Para excluir uma configuração de segurança na região atual

Comando:

```
aws emr delete-security-configuration --name MySecurityConfig
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteSecurityConfiguration](#) na Referência de AWS CLI Comandos.

describe-cluster

O código de exemplo a seguir mostra como usar `describe-cluster`.

AWS CLI

Comando:

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

Saída:

```
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-XXXXXXXX",
    "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-XXXXXXXX",
"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"
}
}

```

- Para API obter detalhes, consulte [DescribeCluster](#) na Referência de AWS CLI Comandos.

describe-step

O código de exemplo a seguir mostra como usar `describe-step`.

AWS CLI

O seguinte comando descreve uma etapa com o ID de etapa `s-3LZC0QUT43AM` em um cluster com o ID de cluster `j-3SD91U2E1L2QX`:

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

Saída:

```

{
  "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"
}
}

```

- Para API obter detalhes, consulte [DescribeStep](#) na Referência de AWS CLI Comandos.

get

O código de exemplo a seguir mostra como usar `get`.

AWS CLI

O seguinte baixa o `hadoop-examples.jar` arquivo da instância principal em um cluster com o ID do cluster `j-3SD91U2E1L2QX`:

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /
home/hadoop-examples.jar --dest ~
```

- Para API obter detalhes, consulte [Get](#) in AWS CLI Command Reference.

list-clusters

O código de exemplo a seguir mostra como usar `list-clusters`.

AWS CLI

O comando a seguir lista todos os EMR clusters ativos na região atual:

```
aws emr list-clusters --active
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListClusters](#) na Referência de AWS CLI Comandos.

list-instance-fleets

O código de exemplo a seguir mostra como usar `list-instance-fleets`.

AWS CLI

Para obter detalhes de configuração das frotas de instâncias em um cluster

Este exemplo lista os detalhes das frotas de instâncias no cluster especificado.

Comando:

```
list-instance-fleets --cluster-id 'j-12ABCDEFGH134JK'
```

Saída:

```
{
  "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"
}
]
}

```

- Para API obter detalhes, consulte [ListInstanceFleets](#) na Referência de AWS CLI Comandos.

list-instances

O código de exemplo a seguir mostra como usar `list-instances`.

AWS CLI

O comando a seguir lista todas as instâncias em um cluster com o ID do `clusterj-3C6XNQ39VR9WL`:

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

Saída:

```

For a uniform instance group based cluster
{
  "Instances": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1433200400.03,

```

```
        "CreationDateTime": 1433199960.152
      },
      "State": "RUNNING",
      "StateChangeReason": {}
    },
    "Ec2InstanceId": "i-f19ecfee",
    "PublicDnsName": "ec2-52-52-41-150.us-west-2.compute.amazonaws.com",
    "PrivateDnsName": "ip-172-21-11-216.us-west-2.compute.internal",
    "PublicIpAddress": "52.52.41.150",
    "Id": "ci-3NNHQUQ2TWB6Y",
    "PrivateIpAddress": "172.21.11.216"
  },
  {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1433200400.031,
        "CreationDateTime": 1433199949.102
      },
      "State": "RUNNING",
      "StateChangeReason": {}
    },
    "Ec2InstanceId": "i-1feee4c2",
    "PublicDnsName": "ec2-52-63-246-32.us-west-2.compute.amazonaws.com",
    "PrivateDnsName": "ip-172-31-24-130.us-west-2.compute.internal",
    "PublicIpAddress": "52.63.246.32",
    "Id": "ci-GAOCMKNKDCV7",
    "PrivateIpAddress": "172.21.11.215"
  },
  {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1433200400.031,
        "CreationDateTime": 1433199949.102
      },
      "State": "RUNNING",
      "StateChangeReason": {}
    },
    "Ec2InstanceId": "i-15cfeee3",
    "PublicDnsName": "ec2-52-25-246-63.us-west-2.compute.amazonaws.com",
    "PrivateDnsName": "ip-172-31-24-129.us-west-2.compute.internal",
    "PublicIpAddress": "52.25.246.63",
    "Id": "ci-2W3TDFFB47UAD",
    "PrivateIpAddress": "172.21.11.214"
  }
}
```



```

    ]
  }

For a fleet based cluster:
{
  "Instances": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1487810810.878,
          "CreationDateTime": 1487810588.367,
          "EndDateTime": 1488022990.924
        },
        "State": "TERMINATED",
        "StateChangeReason": {
          "Message": "Instance was terminated."
        }
      },
      "Ec2InstanceId": "i-xxxxx",
      "InstanceFleetId": "if-xxxxx",
      "EbsVolumes": [],
      "PublicDnsName": "ec2-xx-xxx-xxx-xxx.compute-1.amazonaws.com",
      "InstanceType": "m3.xlarge",
      "PrivateDnsName": "ip-xx-xx-xxx-xx.ec2.internal",
      "Market": "SPOT",
      "PublicIpAddress": "xx.xx.xxx.xxx",
      "Id": "ci-xxxxx",
      "PrivateIpAddress": "10.47.191.80"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListInstances](#) na Referência de AWS CLI Comandos.

list-security-configurations

O código de exemplo a seguir mostra como usar `list-security-configurations`.

AWS CLI

Para listar as configurações de segurança na região atual

Comando:

```
aws emr list-security-configurations
```

Saída:

```
{
  "SecurityConfigurations": [
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-1"
    },
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-2"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListSecurityConfigurations](#) na Referência de AWS CLI Comandos.

list-steps

O código de exemplo a seguir mostra como usar `list-steps`.

AWS CLI

O seguinte comando lista todas as etapas em um cluster com o ID de cluster `j-3SD91U2E1L2QX`:

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- Para API obter detalhes, consulte [ListSteps](#) na Referência de AWS CLI Comandos.

modify-cluster-attributes

O código de exemplo a seguir mostra como usar `modify-cluster-attributes`.

AWS CLI

O comando a seguir define a visibilidade de um EMR cluster com o ID `j-301CDNY0J5XM4` para todos os usuários:

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-users
```

- Para API obter detalhes, consulte [ModifyClusterAttributes](#) na Referência de AWS CLI Comandos.

modify-instance-fleet

O código de exemplo a seguir mostra como usar `modify-instance-fleet`.

AWS CLI

Para alterar as capacidades desejadas de uma frota de instâncias

Este exemplo altera as capacidades alvo sob demanda e spot para 1 para a frota de instâncias especificada.

Comando:

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGH134JK' --instance-fleet InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- Para API obter detalhes, consulte [ModifyInstanceFleet](#) na Referência de AWS CLI Comandos.

put

O código de exemplo a seguir mostra como usar `put`.

AWS CLI

O comando a seguir carrega um arquivo nomeado `healthcheck.sh` para a instância principal em um cluster com o ID do 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
```

- Para API obter detalhes, consulte [Put](#) in AWS CLI Command Reference.

remove-tags

O código de exemplo a seguir mostra como usar `remove-tags`.

AWS CLI

O comando a seguir remove uma tag com a chave `prod` de um cluster com o ID do `clusterj-3SD91U2E1L2QX`:

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- Para API obter detalhes, consulte [RemoveTags](#) na Referência de AWS CLI Comandos.

schedule-hbase-backup

O código de exemplo a seguir mostra como usar `schedule-hbase-backup`.

AWS CLI

Nota: Este comando só pode ser usado HBase nas AMI versões 2.x e 3.x

1. Para agendar um 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
```

Saída:

```
None
```

2. Para agendar um backup incremental 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
```

Saída:

```
None
```

- Para API obter detalhes, consulte [ScheduleHbaseBackup](#) na Referência de AWS CLI Comandos.

socks

O código de exemplo a seguir mostra como usar socks.

AWS CLI

O comando a seguir abre uma conexão socks com a instância mestre em um cluster com o ID do cluster: j-3SD91U2E1L2QX

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

A opção key pair file usa um caminho local para um arquivo de chave privada.

- Para API obter detalhes, consulte [Socks](#) in AWS CLI Command Reference.

ssh

O código de exemplo a seguir mostra como usar ssh.

AWS CLI

O comando a seguir abre uma conexão ssh com a instância principal em um cluster com o ID do cluster: j-3SD91U2E1L2QX

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

A opção key pair file usa um caminho local para um arquivo de chave privada.

Saída:

```
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 ~]$
```

- Para API obter detalhes, consulte [Ssh](#) na Referência de AWS CLI Comandos.

Amazon EMR em EKS exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Amazon EMR onEKS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

update-role-trust-policy

O código de exemplo a seguir mostra como usar `update-role-trust-policy`.

AWS CLI

Para atualizar a política de confiança de uma IAM função a ser usada com a Amazon EMR em EKS

Esse comando de exemplo atualiza a política de confiança de uma função chamada `example_iam_role` para que ela possa ser usada com a Amazon EMR ou com o namespace `example_namespace` de um cluster chamado `example_cluster`. EKS EKS

Comando:

```
aws emr-containers update-role-trust-policy \  
  --cluster example_cluster \  
  --namespace example_namespace \  
  --role-name example_iam_role
```

Saída:

```
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.
```

- Para API obter detalhes, consulte [UpdateRoleTrustPolicy](#) na Referência de AWS CLI Comandos.

EventBridge exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with EventBridge.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-rule

O código de exemplo a seguir mostra como usar `delete-rule`.

AWS CLI

Para excluir uma regra de CloudWatch eventos

Este exemplo exclui a regra chamada `EC2InstanceStateChanges`:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Para API obter detalhes, consulte [DeleteRule](#) na Referência de AWS CLI Comandos.

describe-rule

O código de exemplo a seguir mostra como usar `describe-rule`.

AWS CLI

Para exibir informações sobre uma regra de CloudWatch Eventos

Este exemplo exibe informações sobre a regra chamada `DailyLambdaFunction`:


```
aws events describe-rule --name "DailyLambdaFunction"
```

- Para API obter detalhes, consulte [DescribeRule](#) na Referência de AWS CLI Comandos.

disable-rule

O código de exemplo a seguir mostra como usar `disable-rule`.

AWS CLI

Para desativar uma regra de CloudWatch eventos

Este exemplo desativa a regra chamada `DailyLambdaFunction`. A regra não é excluída:

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Para API obter detalhes, consulte [DisableRule](#) na Referência de AWS CLI Comandos.

enable-rule

O código de exemplo a seguir mostra como usar `enable-rule`.

AWS CLI

Para habilitar uma regra de CloudWatch Eventos

Este exemplo ativa a regra chamada `DailyLambdaFunction`, que havia sido desativada anteriormente:

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Para API obter detalhes, consulte [EnableRule](#) na Referência de AWS CLI Comandos.

list-rule-names-by-target

O código de exemplo a seguir mostra como usar `list-rule-names-by-target`.

AWS CLI

Como exibir todas as regras que têm um destino especificado

Este exemplo exibe todas as regras que têm a função Lambda chamada "MyFunctionName" como destino:

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

- Para API obter detalhes, consulte [ListRuleNamesByTarget](#) na Referência de AWS CLI Comandos.

list-rules

O código de exemplo a seguir mostra como usar `list-rules`.

AWS CLI

Para exibir uma lista de todas as regras de CloudWatch eventos

Este exemplo exibe todas as regras de CloudWatch eventos na região:

```
aws events list-rules
```

Para exibir uma lista de regras de CloudWatch eventos começando com uma determinada string.

Este exemplo exibe todas as regras de CloudWatch eventos na região que têm um nome começando com "Diário":

```
aws events list-rules --name-prefix "Daily"
```

- Para API obter detalhes, consulte [ListRules](#) na Referência de AWS CLI Comandos.

list-targets-by-rule

O código de exemplo a seguir mostra como usar `list-targets-by-rule`.

AWS CLI

Para exibir todos os alvos de uma regra de CloudWatch Eventos

Este exemplo exibe todos os alvos da regra chamada DailyLambdaFunction:

```
aws events list-targets-by-rule --rule "DailyLambdaFunction"
```

- Para API obter detalhes, consulte [ListTargetsByRule](#) na Referência de AWS CLI Comandos.

put-events

O código de exemplo a seguir mostra como usar `put-events`.

AWS CLI

Para enviar um evento personalizado para CloudWatch Eventos

Este exemplo envia um evento personalizado para CloudWatch Eventos. O evento está contido no arquivo `putevents.json`:

```
aws events put-events --entries file://putevents.json
```

Veja a seguir o conteúdo do arquivo `putevent.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"
  }
]
```

- Para API obter detalhes, consulte [PutEvents](#) na Referência de AWS CLI Comandos.

put-rule

O código de exemplo a seguir mostra como usar `put-rule`.

AWS CLI

Para criar regras de CloudWatch eventos

Este exemplo cria uma regra que é acionada todos os dias às 9:00 da manhã (). UTC Se você usar `put-targets` para adicionar uma função do Lambda como destino dessa regra, poderá executar a função do Lambda todos os dias no horário especificado:

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 * * ? *)"
```

Este exemplo cria uma regra que é acionada quando qualquer EC2 instância na região muda de estado:

```
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"
```

Este exemplo cria uma regra que é acionada quando qualquer EC2 instância na região é interrompida ou encerrada:

```
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"
```

- Para API obter detalhes, consulte [PutRule](#) na Referência de AWS CLI Comandos.

put-targets

O código de exemplo a seguir mostra como usar `put-targets`.

AWS CLI

Para adicionar alvos às regras de CloudWatch eventos

Este exemplo adiciona uma função do Lambda como o destino de uma regra:

```
aws events put-targets --rule DailyLambdaFunction --targets  
"Id"="1", "Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Este exemplo define um fluxo do Amazon Kinesis como o destino. Desta forma, os eventos capturados pela regra são retransmitidos para o fluxo:

```
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"
```

Este exemplo define dois fluxos do Amazon Kinesis como destinos de uma regra:

```
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"
```

- Para API obter detalhes, consulte [PutTargets](#) na Referência de AWS CLI Comandos.

remove-targets

O código de exemplo a seguir mostra como usar `remove-targets`.

AWS CLI

Como remover um destino de um evento

Este exemplo remove o stream do Amazon Kinesis chamado MyStream 1 de ser um alvo da regra. `DailyLambdaFunction` Quando `DailyLambdaFunction` foi criado, esse fluxo foi definido como um destino com um ID de `Target1`:

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Para API obter detalhes, consulte [RemoveTargets](#) na Referência de AWS CLI Comandos.

test-event-pattern

O código de exemplo a seguir mostra como usar `test-event-pattern`.

AWS CLI

Para verificar se um padrão de evento corresponde a um evento especificado

Este exemplo testa se o padrão “source:com.mycompany.myapp” corresponde ao evento especificado. Neste exemplo, a saída seria “verdadeira”:

```
aws events test-event-pattern --event-pattern "{\"source\": [\"com.mycompany.myapp
\"]}" --event "{\"id\": \"1\", \"source\": \"com.mycompany.myapp\", \"detail-type\":
\"myDetailType\", \"account\": \"123456789012\", \"region\": \"us-east-1\", \"time\":
\"2017-04-11T20:11:04Z\"}"
```

- Para API obter detalhes, consulte [TestEventPattern](#) na Referência de AWS CLI Comandos.

Exemplos do Firewall Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Firewall Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-admin-account

O código de exemplo a seguir mostra como usar `associate-admin-account`.

AWS CLI

Para definir a conta de administrador do Firewall Manager

O `associate-admin-account` exemplo a seguir define a conta de administrador do Firewall Manager.

```
aws fms associate-admin-account \  
  --admin-account 123456789012
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurar a conta de administrador do AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [AssociateAdminAccount](#) na Referência de AWS CLI Comandos.

delete-notification-channel

O código de exemplo a seguir mostra como usar `delete-notification-channel`.

AWS CLI

Para remover as informações do SNS tópico dos registros do Firewall Manager

O `delete-notification-channel` exemplo a seguir remove as informações do SNS tópico.

```
aws fms delete-notification-channel
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurar SNS notificações da Amazon e CloudWatch alarmes da Amazon](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeleteNotificationChannel](#) na Referência de AWS CLI Comandos.

delete-policy

O código de exemplo a seguir mostra como usar `delete-policy`.

AWS CLI

Para excluir uma política do Firewall Manager

O `delete-policy` exemplo a seguir remove a política com o ID especificado, junto com todos os seus recursos.

```
aws fms delete-policy \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --delete-all-policy-resources
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com políticas do AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

disassociate-admin-account

O código de exemplo a seguir mostra como usar `disassociate-admin-account`.

AWS CLI

Para remover a conta de administrador do Firewall Manager

O `disassociate-admin-account` exemplo a seguir remove a associação atual da conta de administrador do Firewall Manager.

```
aws fms disassociate-admin-account
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurar a conta de administrador do AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DisassociateAdminAccount](#) na Referência de AWS CLI Comandos.

get-admin-account

O código de exemplo a seguir mostra como usar `get-admin-account`.

AWS CLI

Para recuperar a conta de administrador do Firewall Manager

O `get-admin-account` exemplo a seguir recupera a conta do administrador.


```
aws fms get-admin-account
```

Saída:

```
{
  "AdminAccount": "123456789012",
  "RoleStatus": "READY"
}
```

Para obter mais informações, consulte [Pré-requisitos do AWS Firewall Manager](#) no AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetAdminAccount](#) na Referência de AWS CLI Comandos.

get-compliance-detail

O código de exemplo a seguir mostra como usar `get-compliance-detail`.

AWS CLI

Para recuperar as informações de conformidade de uma conta

O `get-compliance-detail` exemplo a seguir recupera as informações de conformidade da política e da conta do membro especificadas.

```
aws fms get-compliance-detail \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --member-account 123456789012
```

Saída:

```
{
  "PolicyComplianceDetail": {
    "EvaluationLimitExceeded": false,
    "IssueInfoMap": {},
    "MemberAccount": "123456789012",
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyOwner": "123456789012",
    "Violators": []
  }
}
```

Para obter mais informações, consulte [Exibindo a conformidade de recursos com uma política](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetComplianceDetail](#) na Referência de AWS CLI Comandos.

get-notification-channel

O código de exemplo a seguir mostra como usar `get-notification-channel`.

AWS CLI

Para recuperar as informações do SNS tópico para os registros do Firewall Manager

O `get-notification-channel` exemplo a seguir recupera as informações do SNS tópico.

```
aws fms get-notification-channel
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Configurar SNS notificações da Amazon e CloudWatch alarmes da Amazon](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetNotificationChannel](#) na Referência de AWS CLI Comandos.

get-policy

O código de exemplo a seguir mostra como usar `get-policy`.

AWS CLI

Para recuperar uma política do Firewall Manager

O `get-policy` exemplo a seguir recupera a política com o ID especificado.

```
aws fms get-policy \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Trabalhando com políticas do AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetPolicy](#) na Referência de AWS CLI Comandos.

list-compliance-status

O código de exemplo a seguir mostra como usar `list-compliance-status`.

AWS CLI

Para recuperar as informações de conformidade com a política das contas dos membros

O `list-compliance-status` exemplo a seguir recupera as informações de conformidade da conta do membro para a política especificada.

```
aws fms list-compliance-status \
```

```
--policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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": {}
    }
  ]
}
```

Para obter mais informações, consulte [Exibindo a conformidade de recursos com uma política](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListComplianceStatus](#) na Referência de AWS CLI Comandos.

list-member-accounts

O código de exemplo a seguir mostra como usar `list-member-accounts`.

AWS CLI

Para recuperar as contas dos membros na organização

O `list-member-accounts` exemplo a seguir lista todas as contas membros que estão na organização do administrador do Firewall Manager.

```
aws fms list-member-accounts
```

Saída:

```
{
  "MemberAccounts": [
    "222222222222",
    "333333333333",
    "444444444444"
  ]
}
```

Para obter mais informações, consulte [AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListMemberAccounts](#) na Referência de AWS CLI Comandos.

list-policies

O código de exemplo a seguir mostra como usar `list-policies`.

AWS CLI

Para recuperar todas as políticas do Firewall Manager

O `list-policies` exemplo a seguir recupera a lista de políticas da conta. Neste exemplo, a saída é limitada a dois resultados por solicitação. Cada chamada retorna um `NextToken` que pode ser usado como o valor do `--starting-token` parâmetro na próxima `list-policies` chamada para obter o próximo conjunto de resultados para a lista.

```
aws fms list-policies \
  --max-items 2
```

Saída:

```
{
  "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=="
}

```

Para obter mais informações, consulte [Trabalhando com políticas do AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListPolicies](#) na Referência de AWS CLI Comandos.

put-notification-channel

O código de exemplo a seguir mostra como usar `put-notification-channel`.

AWS CLI

Para definir as informações do SNS tópico para os registros do Firewall Manager

O `put-notification-channel` exemplo a seguir define as informações do SNS tópico.

```

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

```

Este comando não produz saída.

Para obter mais informações, consulte [Configurar SNS notificações da Amazon e CloudWatch alarmes da Amazon](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [PutNotificationChannel](#) na Referência de AWS CLI Comandos.

put-policy

O código de exemplo a seguir mostra como usar `put-policy`.

AWS CLI

Para criar uma política do Firewall Manager

O `put-policy` exemplo a seguir cria uma política de grupo de segurança do Firewall Manager.

```
aws fms put-policy \  
  --cli-input-json file://policy.json
```

Conteúdo de `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"  
    }  
  ]  
}
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Trabalhando com políticas do AWS Firewall Manager](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [PutPolicy](#) na Referência de AWS CLI Comandos.

AWS FIS exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS FIS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-experiment-template

O código de exemplo a seguir mostra como usar `create-experiment-template`.

AWS CLI

Para criar um modelo de experimento

O `create-experiment-template` exemplo a seguir cria um modelo de experimento em sua AWS FIS conta.

```
aws fis create-experiment-template \  
  --cli-input-json file://myfile.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Criar um modelo de experimento](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [CreateExperimentTemplate](#) na Referência de AWS CLI Comandos.

delete-experiment-template

O código de exemplo a seguir mostra como usar delete-experiment-template.

AWS CLI

Para excluir um modelo de experimento

O delete-experiment-template exemplo a seguir exclui o modelo de experimento especificado.

```
aws fis delete-experiment-template \  
  --id ABCDE1fgHIJKLmNop
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Excluir um modelo de experimento](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [DeleteExperimentTemplate](#) na Referência de AWS CLI Comandos.

get-action

O código de exemplo a seguir mostra como usar `get-action`.

AWS CLI

Para obter detalhes da ação

O `get-action` exemplo a seguir obtém os detalhes da ação especificada.

```
aws fis get-action \
  --id aws:ec2:stop-instances
```

Saída:

```
{
  "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": {}
}
}

```

Para obter mais informações, consulte [Ações](#) no Guia do usuário do simulador de injeção de AWS falhas.

- Para API obter detalhes, consulte [GetAction](#) na Referência de AWS CLI Comandos.

get-experiment-template

O código de exemplo a seguir mostra como usar `get-experiment-template`.

AWS CLI

Para obter detalhes do modelo de experimento

O `get-experiment-template` exemplo a seguir obtém os detalhes do modelo de experimento especificado.

```

aws fis get-experiment-template \
  --id ABCDE1fgHIJKLmNop

```

Saída:

```

{
  "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"
}
}
```

Para obter mais informações, consulte [Modelos de experimentos](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [GetExperimentTemplate](#) na Referência de AWS CLI Comandos.

get-experiment

O código de exemplo a seguir mostra como usar `get-experiment`.

AWS CLI

Para obter detalhes do experimento

O `get-experiment` exemplo a seguir obtém os detalhes do experimento especificado.

```
aws fis get-experiment \  
--id ABC12DeFGhI3jKLMNOP
```

Saída:

```
{  
  "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": {}
  }
}
```

Para obter mais informações, consulte [Experimentos AWS FIS](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [GetExperiment](#) na Referência de AWS CLI Comandos.

list-actions

O código de exemplo a seguir mostra como usar `list-actions`.

AWS CLI

Para listar ações

O `list-actions` exemplo a seguir lista as ações disponíveis.

```
aws fis list-actions
```

Saída:

```
{
  "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": {}
  }
]
}

```

Para obter mais informações, consulte [Ações](#) no Guia do usuário do simulador de injeção de AWS falhas.

- Para API obter detalhes, consulte [ListActions](#) na Referência de AWS CLI Comandos.

list-experiment-templates

O código de exemplo a seguir mostra como usar `list-experiment-templates`.

AWS CLI

Para listar modelos de experimentos

O `list-experiment-templates` exemplo a seguir lista os modelos de experimentos em sua AWS conta.

```
aws fis list-experiment-templates
```

Saída:

```
{
```

```
"experimentTemplates": [  
  {  
    "id": "ABCDE1fgHIJkLmNop",  
    "description": "myExperimentTemplate",  
    "creationTime": 1616017191.124,  
    "lastUpdateTime": 1616017191.124,  
    "tags": {  
      "key": "value"  
    }  
  }  
]
```

Para obter mais informações, consulte [Modelos de experimentos](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [ListExperimentTemplates](#) na Referência de AWS CLI Comandos.

list-experiments

O código de exemplo a seguir mostra como usar `list-experiments`.

AWS CLI

Para listar experimentos

O `list-experiments` exemplo a seguir lista os experimentos em sua AWS conta.

```
aws fis list-experiments
```

Saída:

```
{  
  "experiments": [  
    {  
      "id": "ABCdeF1GHiJkLM23N0",  
      "experimentTemplateId": "ABCDE1fgHIJkLmNop",  
      "state": {  
        "status": "running",  
        "reason": "Experiment is running."  
      },  
      "creationTime": 1616017341.197,  
    }  
  ]  
}
```

```
        "tags": {
          "key": "value"
        }
      ]
    }
  }
```

Para obter mais informações, consulte [Experimentos](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [ListExperiments](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags do recurso especificado.

```
aws fis list-tags-for-resource \
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Saída:

```
{
  "tags": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Para obter mais informações, consulte [Marcar seus AWS FIS recursos](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

start-experiment

O código de exemplo a seguir mostra como usar `start-experiment`.

AWS CLI

Para começar um experimento

O `start-experiment` exemplo a seguir inicia o experimento especificado.

```
aws fis start-experiment \  
  --experiment-template-id ABCDE1fgHIJKLmNop
```

Saída:

```
{  
  "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": {}
  }
}
```

Para obter mais informações, consulte [Experimentos AWS FIS](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [StartExperiment](#) na Referência de AWS CLI Comandos.

stop-experiment

O código de exemplo a seguir mostra como usar stop-experiment.

AWS CLI

Para interromper um experimento

O stop-experiment exemplo a seguir interrompe a execução do experimento especificado.

```
aws fis stop-experiment \
  --id ABC12DeFGhI3jKLMNOP
```

Saída:

```
{
  "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": {}
}
```


Para obter mais informações, consulte [Experimentos AWS FIS](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [StopExperiment](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir marca o recurso especificado.

```
aws fis tag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP \  
  --tags key1=value1,key2=value2
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar seus AWS FIS recursos](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para desmarcar um recurso

O `untag-resource` exemplo a seguir remove as tags do recurso especificado.

```
aws fis untag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar seus AWS FIS recursos](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-experiment-template

O código de exemplo a seguir mostra como usar `update-experiment-template`.

AWS CLI

Para atualizar um modelo de experimento

O `update-experiment-template` exemplo a seguir atualiza a descrição do modelo de experimento especificado.

```
aws fis update-experiment-template \  
  --id ABCDE1fgHIJkLmNop \  
  ---description myExperimentTemplate
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Atualizar um modelo de experimento](#) no Guia do usuário do AWS Fault Injection Simulator.

- Para API obter detalhes, consulte [UpdateExperimentTemplate](#) na Referência de AWS CLI Comandos.

GameLift Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon GameLift.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-build

O código de exemplo a seguir mostra como usar `create-build`.

AWS CLI

Exemplo 1: Para criar uma compilação de jogo a partir de arquivos em um bucket do S3

O `create-build` exemplo a seguir cria um recurso personalizado de criação de jogos. Ele usa arquivos compactados que são armazenados em um local do S3 em uma AWS conta que você controla. Este exemplo pressupõe que você já tenha criado uma IAM função que dá GameLift permissão à Amazon para acessar a localização do S3. Como a solicitação não especifica um sistema operacional, o novo recurso de compilação usa como padrão `_2012.WINDOWS`

```
aws gamelift create-build \  
  --storage-location file://storage-loc.json \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678
```

Conteúdo de `storage-loc.json`:

```
{  
  "Bucket": "MegaFrogRaceServer_NA_build_files"  
  "Key": "MegaFrogRaceServer_build_123.zip"  
  "RoleArn": "arn:aws:iam::123456789012:role/gamelift"  
}
```

Saída:

```
{  
  "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"
  }
}

```

Exemplo 2: Para criar um recurso de criação de jogos para fazer upload manual de arquivos para GameLift

O `create-build` exemplo a seguir cria um novo recurso de compilação. Ele também obtém um local de armazenamento e credenciais temporárias que permitem que você faça o upload manual da compilação do jogo para o GameLift local no Amazon S3. Depois de carregar sua compilação com sucesso, o GameLift serviço valida a compilação e atualiza o status da nova compilação.

```

aws gamelift create-build \
  --name MegaFrogRaceServer.NA \
  --build-version 12345.678 \
  --operating-system AMAZON_LINUX

```

Saída:

```

{
  "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": "AgoGb3JpZ21uENZ...EXAMPLETOKEN=="
  }
}

```

Para obter mais informações, consulte [Fazer upload de uma compilação de servidor personalizada GameLift](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [CreateBuild](#) na Referência de AWS CLI Comandos.

create-fleet

O código de exemplo a seguir mostra como usar `create-fleet`.

AWS CLI

Exemplo 1: Para criar uma frota Linux básica

O `create-fleet` exemplo a seguir cria uma frota minimamente configurada de instâncias Linux sob demanda para hospedar uma criação de servidor personalizada. Você pode concluir a configuração usando `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}]'

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 2: Para criar uma frota básica do Windows

O `create-fleet` exemplo a seguir cria uma frota minimamente configurada de instâncias spot do Windows para hospedar uma criação de servidor personalizada. Você pode concluir a configuração usando `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}]'

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 3: Para criar uma frota totalmente configurada

O `create-fleet` exemplo a seguir cria uma frota de instâncias Spot do Windows para uma criação de servidor personalizada, com as configurações mais usadas fornecidas.

```

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'

```

Conteúdo de `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}]

```

Saída:


```
{
  "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"
    }
  }
}
```

Exemplo 4: Para criar uma frota de servidores em tempo real

O `create-fleet` exemplo a seguir cria uma frota de instâncias spot com um script de configuração em tempo real que foi enviado para a Amazon GameLift. Todos os servidores em tempo real são implantados em máquinas Linux. Para fins deste exemplo, suponha que o script em tempo real carregado inclua vários arquivos de script, com a `Init()` função localizada no arquivo de script chamada `MainScript.js`. Conforme mostrado, esse arquivo é identificado como o script de inicialização na configuração do tempo de execução.

```
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}]'
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateFleet](#) na Referência de AWS CLI Comandos.

create-game-session-queue

O código de exemplo a seguir mostra como usar create-game-session-queue.

AWS CLI

Exemplo 1: Para configurar uma fila ordenada de sessões de jogo

O create-game-session-queue exemplo a seguir cria uma nova fila de sessões de jogo com destinos em duas regiões. Ele também configura a fila para que as solicitações da sessão do

jogo expirem após esperar 10 minutos pela colocação. Como nenhuma política de latência está definida, GameLift tenta colocar todas as sessões de jogo com o primeiro destino listado.

```
aws gamelift create-game-session-queue \  
  --name MegaFrogRaceServer-NA \  
  --destinations file://destinations.json \  
  --timeout-in-seconds 600
```

Conteúdo de `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"}  
  ]  
}
```

Saída:

```
{  
  "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"}  
      ]  
    }  
  ]  
}
```

Exemplo 2: Para configurar uma fila de sessões de jogo com políticas de latência do jogador

O `create-game-session-queue` exemplo a seguir cria uma nova fila de sessões de jogo com políticas de latência para dois jogadores. A primeira política define um limite de latência

de 100 ms que é aplicado durante o primeiro minuto de uma tentativa de colocação na sessão de jogo. A segunda política aumenta o limite de latência para 200 ms até que a solicitação de posicionamento expire em 3 minutos.

```
aws gamelift create-game-session-queue \  
  --name MegaFrogRaceServer-NA \  
  --destinations file://destinations.json \  
  --player-latency-policies file://latency-policies.json \  
  --timeout-in-seconds 180
```

Conteúdo de `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" }  
  ]  
}
```

Conteúdo de `latency-policies.json`:

```
{  
  "PlayerLatencyPolicies": [  
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},  
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":  
60}  
  ]  
}
```

Saída:

```
{  
  "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"}
    ],
  }
}
```

Para obter mais informações, consulte [Criar uma fila](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [CreateGameSessionQueue](#) na Referência de AWS CLI Comandos.

delete-build

O código de exemplo a seguir mostra como usar delete-build.

AWS CLI

Para excluir uma versão de jogo personalizada

O delete-build exemplo a seguir remove uma compilação da sua GameLift conta da Amazon. Depois que a compilação for excluída, você não poderá usá-la para criar novas frotas. Essa operação não pode ser desfeita.

```
aws gamelift delete-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBuild](#) na Referência de AWS CLI Comandos.

delete-fleet

O código de exemplo a seguir mostra como usar delete-fleet.

AWS CLI

Para excluir uma frota que não está mais em uso

O `delete-fleet` exemplo a seguir remove uma frota que foi reduzida para zero instâncias. Se a capacidade da frota for maior que zero, a solicitação falhará com um erro HTTP 400.

```
aws gamelift delete-fleet \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar GameLift frotas](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DeleteFleet](#) na Referência de AWS CLI Comandos.

`delete-game-session-queue`

O código de exemplo a seguir mostra como usar `delete-game-session-queue`.

AWS CLI

Para excluir uma fila de sessões de jogo

O `delete-game-session-queue` exemplo a seguir exclui uma fila de sessão de jogo especificada.

```
aws gamelift delete-game-session-queue \  
  --name MegaFrogRace-NA
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteGameSessionQueue](#) na Referência de AWS CLI Comandos.

`describe-build`

O código de exemplo a seguir mostra como usar `describe-build`.

AWS CLI

Para obter informações sobre uma versão de jogo personalizada

O `describe-build` exemplo a seguir recupera propriedades de um recurso de criação de servidor de jogos.

```
aws gamelift describe-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Fazer upload de uma compilação de servidor personalizada GameLift](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeBuild](#) na Referência de AWS CLI Comandos.

describe-ec2-instance-limits

O código de exemplo a seguir mostra como usar `describe-ec2-instance-limits`.

AWS CLI

Para recuperar os limites de serviço para um tipo de EC2 instância

O `describe-ec2-instance-limits` exemplo a seguir mostra o máximo permitido de instâncias e instâncias atuais em uso para o tipo de EC2 instância especificado na região atual. O resultado indica que apenas cinco das vinte instâncias permitidas estão sendo usadas.

```
aws gamelift describe-ec2-instance-limits \  
  --ec2-instance-type m5.Large
```

Saída:

```
{
  "EC2InstanceLimits": [
    {
      "EC2InstanceType": "m5.large",
      "CurrentInstances": 5,
      "InstanceLimit": 20
    }
  ]
}
```

Para obter mais informações, consulte [Choose Computing Resources](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeEc2 InstanceLimits](#) em Referência de AWS CLI comandos.

describe-fleet-attributes

O código de exemplo a seguir mostra como usar `describe-fleet-attributes`.

AWS CLI

Exemplo 1: Para visualizar os atributos de uma lista de frotas

O `describe-fleet-attributes` exemplo a seguir recupera os atributos da frota de duas frotas especificadas. Conforme mostrado, as frotas solicitadas são implantadas com a mesma construção, uma para instâncias sob demanda e outra para instâncias spot, com algumas pequenas diferenças de configuração.

```
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
```

Saída:

```
{
  "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"
    }
}
```

```

    }
  ]
}

```

Exemplo 2: Para solicitar atributos para todas as frotas

O seguinte `describe-fleet-attributes` retorna os atributos da frota para todas as frotas com qualquer status. Este exemplo ilustra o uso de parâmetros de paginação para retornar uma frota por vez.

```

aws gamelift describe-fleet-attributes \
  --limit 1

```

Saída:

```

{
  "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":
    "eyJhd3NBZjY2NvdW50SWQlOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"

```

```
}

```

A saída inclui um NextToken valor que você pode usar ao chamar o comando pela segunda vez. Passe o valor para o `--next-token` parâmetro para especificar onde obter a saída. O comando a seguir retorna o segundo resultado na saída.

```
aws gamelift describe-fleet-attributes \
  --limit 1 \
  --next-
token eyJhd3NBY2NvdW50SWQi0nsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Repita até que a resposta não inclua um NextToken valor.

Para obter mais informações, consulte [Configurando GameLift frotas](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeFleetAttributes](#) na Referência de AWS CLI Comandos.

describe-fleet-capacity

O código de exemplo a seguir mostra como usar `describe-fleet-capacity`.

AWS CLI

Para ver o status da capacidade de uma lista de frotas

O `describe-fleet-capacity` exemplo a seguir recupera a capacidade atual de duas frotas especificadas.

```
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
```

Saída:

```
{
  "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
    }
  }
]
}
```

Para obter mais informações, consulte [GameLift Métricas para frotas](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeFleetCapacity](#) na Referência de AWS CLI Comandos.

describe-fleet-events

O código de exemplo a seguir mostra como usar `describe-fleet-events`.

AWS CLI

Para solicitar eventos para um período de tempo especificado

O `describe-fleet-events` exemplo a seguir exibe detalhes de todos os eventos relacionados à frota que ocorreram durante o período especificado.

```
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
```

Saída:

```
{  
  "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  
    },  
  ]  
}
```


AWS CLI

Para ver as permissões de conexão de entrada para uma frota

O `describe-fleet-port-settings` exemplo a seguir recupera as configurações de conexão de uma frota especificada.

```
aws gamelift describe-fleet-port-settings \  
  --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Configurando GameLift frotas](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeFleetPortSettings](#) na Referência de AWS CLI Comandos.

describe-fleet-utilization

O código de exemplo a seguir mostra como usar `describe-fleet-utilization`.

AWS CLI

Exemplo 1: Para visualizar os dados de uso de uma lista de frotas

O `describe-fleet-utilization` exemplo a seguir recupera as informações de uso atuais de uma frota especificada.

```
aws gamelift describe-fleet-utilization \  
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111
```

Saída:

```
{  
  "FleetUtilization": [  
    {  
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "ActiveServerProcessCount": 100,  
      "ActiveGameSessionCount": 62,  
      "CurrentPlayerSessionCount": 329,  
      "MaximumPlayerSessionCount": 1000  
    }  
  ]  
}
```

Exemplo 2: Para solicitar dados de uso para todas as frotas

O seguinte `describe-fleet-utilization` retorna dados de uso da frota para todas as frotas com qualquer status. Este exemplo usa parâmetros de paginação para retornar dados de duas frotas ao mesmo tempo.

```
aws gamelift describe-fleet-utilization \  
  --limit 2
```

Saída:

```
{  
  "FleetUtilization": [  
    {  
      "FleetId": "fleet-1111aaaa-22bb-33cc-44dd-5555eeee66ff",  
      "ActiveServerProcessCount": 100,  
      "ActiveGameSessionCount": 62,  
      "CurrentPlayerSessionCount": 329,  
      "MaximumPlayerSessionCount": 1000  
    }  
  ]  
}
```



```

        "ActiveGameSessionCount": 13,
        "CurrentPlayerSessionCount": 98,
        "MaximumPlayerSessionCount": 1000
    },
    {
        "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
        "ActiveServerProcessCount": 100,
        "ActiveGameSessionCount": 62,
        "CurrentPlayerSessionCount": 329,
        "MaximumPlayerSessionCount": 1000
    }
],
"NextToken":
"eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}

```

Chame o comando pela segunda vez, passando o NextToken valor como argumento para o `--next-token` parâmetro para ver os próximos dois resultados.

```

aws gamelift describe-fleet-utilization \
  --limit 2 \
  --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS

```

Repita até que a resposta não inclua mais um NextToken valor na saída.

Para obter mais informações, consulte [GameLift Métricas para frotas](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeFleetUtilization](#) na Referência de AWS CLI Comandos.

describe-game-session-queues

O código de exemplo a seguir mostra como usar `describe-game-session-queues`.

AWS CLI

Para ver as filas de sessões de jogo

O `describe-game-session-queues` exemplo a seguir recupera propriedades de duas filas especificadas.

```
aws gamelift describe-game-session-queues \
  --names MegaFrogRace-NA MegaFrogRace-EU
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte Como [usar filas multirregionais](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeGameSessionQueues](#) na Referência de AWS CLI Comandos.

describe-runtime-configuration

O código de exemplo a seguir mostra como usar `describe-runtime-configuration`.

AWS CLI

Para solicitar a configuração de tempo de execução de uma frota

O `describe-runtime-configuration` exemplo a seguir recupera detalhes sobre a configuração atual do tempo de execução de uma frota especificada.

```

aws gamelift describe-runtime-configuration \
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "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
}
}
```

Para obter mais informações, consulte [Executar vários processos em uma frota](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [DescribeRuntimeConfiguration](#) na Referência de AWS CLI Comandos.

list-builds

O código de exemplo a seguir mostra como usar `list-builds`.

AWS CLI

Exemplo 1: Para obter uma lista de versões de jogos personalizadas

O `list-builds` exemplo a seguir recupera as propriedades de todas as compilações de servidores de jogos na região atual. O exemplo de solicitação ilustra como usar os parâmetros de paginação `Limit` e recuperar `NextToken` os resultados em conjuntos sequenciais. O primeiro comando recupera as duas primeiras compilações. Como há mais de dois disponíveis, a resposta inclui um `NextToken` para indicar que há mais resultados disponíveis.

```
aws gamelift list-builds \
  --limit 2
```

Saída:

```
{
  "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":
"eyJhd3NBWY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS
}

```

Em seguida, você pode chamar o comando novamente com o `--next-token` parâmetro a seguir para ver as próximas duas compilações.

```

aws gamelift list-builds \
  --limit 2
  --next-
token eyJhd3NBWY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS

```

Repita até que a resposta não inclua um NextToken valor.

Exemplo 2: Para obter uma lista de compilações de jogos personalizadas em status de falha

O `list-builds` exemplo a seguir recupera propriedades de todas as compilações de servidores de jogos na região atual que atualmente têm status. FAILED

```

aws gamelift list-builds \
  --status FAILED

```

Saída:

```
{
```

```

    "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"
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ListBuilds](#) na Referência de AWS CLI Comandos.

list-fleets

O código de exemplo a seguir mostra como usar `list-fleets`.

AWS CLI

Exemplo 1: Para obter uma lista de todas as frotas em uma região

O `list-fleets` exemplo a seguir mostra a frota IDs de todas as frotas na região atual. Este exemplo usa parâmetros de paginação para recuperar duas frotas por IDs vez. A resposta inclui um `next-token` atributo, que indica que há mais resultados a serem recuperados.

```

aws gamelift list-fleets \
  --limit 2

```

Saída:

```

{
  "FleetIds": [
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  ],
  "NextToken":
    "eyJhd3NBY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS"
}

```

```
}

```

Você pode passar o NextToken valor da resposta anterior no próximo comando, conforme mostrado aqui, para obter os próximos dois resultados.

```
aws gamelift list-fleets \
  --limit 2 \
  --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRLZj
```

Exemplo 2: Para obter uma lista de todas as frotas em uma região com uma construção ou script específico

O list-builds exemplo a seguir recupera as IDs frotas implantadas com a compilação de jogo especificada. Se você estiver trabalhando com servidores em tempo real, poderá fornecer uma ID de script no lugar de uma ID de compilação. Como esse exemplo não especifica o parâmetro limite, os resultados podem incluir até 16 frotasIDs.

```
aws gamelift list-fleets \
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "FleetIds": [
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"
  ]
}
```

- Para API obter detalhes, consulte [ListFleets](#) na Referência de AWS CLI Comandos.

request-upload-credentials

O código de exemplo a seguir mostra como usar request-upload-credentials.

AWS CLI

Para atualizar as credenciais de acesso para fazer o upload de uma compilação

O `create-build` exemplo a seguir obtém credenciais de acesso novas e válidas para carregar um arquivo de GameLift compilação em um local do Amazon S3. As credenciais têm uma vida útil limitada. Você obtém o ID da compilação da resposta à `CreateBuild` solicitação original.

```
aws gamelift request-upload-credentials \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "StorageLocation": {  
    "Bucket": "gamelift-builds-us-west-2",  
    "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  },  
  "UploadCredentials": {  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",  
    "SessionToken": "AgoGb3JpZ21uENz...EXAMPLETOKEN=="  
  }  
}
```

Para obter mais informações, consulte [Fazer upload de uma compilação de servidor personalizada GameLift](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [RequestUploadCredentials](#) na Referência de AWS CLI Comandos.

start-fleet-actions

O código de exemplo a seguir mostra como usar `start-fleet-actions`.

AWS CLI

Para reiniciar a atividade de escalonamento automático da frota

O `start-fleet-actions` exemplo a seguir retoma o uso de todas as políticas de escalabilidade definidas para a frota especificada, mas que foram interrompidas `stop-fleet-actions` chamando `` ``. Depois de começar, as políticas de escalabilidade começam imediatamente a monitorar suas respectivas métricas.

```
aws gamelift start-fleet-actions \  
  --fleet-id fleet-123456789012
```



```
--build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
--name MegaFrogRaceServer.NA.east \  
--build-version 12345.east
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Atualizar seus arquivos de compilação](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [UpdateBuild](#) na Referência de AWS CLI Comandos.

update-game-session-queue

O código de exemplo a seguir mostra como usar update-game-session-queue.

AWS CLI

Para atualizar a configuração da fila de uma sessão de jogo

O update-game-session-queue exemplo a seguir adiciona um novo destino e atualiza as políticas de latência do jogador para uma fila de sessão de jogo existente.

```
aws gamelift update-game-session-queue \  
--name MegaFrogRace-NA \  
--destinations file://destinations.json \  
--player-latency-policies file://latency-policies.json
```

Conteúdo de 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"}
  ]
}
```

Conteúdo de `latency-policies.json`:

```
{
  "PlayerLatencyPolicies": [
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},
    {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
120},
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
120}
  ]
}
```

Saída:

```
{
  "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}
    ]
}
}

```

Para obter mais informações, consulte Como [usar filas multirregionais](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [UpdateGameSessionQueue](#) na Referência de AWS CLI Comandos.

upload-build

O código de exemplo a seguir mostra como usar upload-build.

AWS CLI

Exemplo 1: Para fazer o upload de uma compilação de servidor de jogos Linux

O upload-build exemplo a seguir carrega arquivos de compilação do servidor de jogos Linux de um diretório de arquivos para o GameLift serviço e cria um recurso de compilação.

```

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

```

Saída:

```

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

```

Exemplo 2: Para carregar uma compilação de servidor de jogos do Windows

O upload-build exemplo a seguir carrega arquivos de compilação do servidor de jogos Windows de um diretório para o GameLift serviço e cria um registro de compilação.

```
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
```

Saída:

```
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
```

Para obter mais informações, consulte [Fazer upload de uma compilação de servidor personalizada GameLift](#) no Amazon GameLift Developer Guide.

- Para API obter detalhes, consulte [UploadBuild](#) na Referência de AWS CLI Comandos.

Exemplos do Global Accelerator usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Global Accelerator.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-custom-routing-endpoints

O código de exemplo a seguir mostra como usar `add-custom-routing-endpoints`.

AWS CLI

Para adicionar um endpoint de VPC sub-rede a um grupo de endpoints para um acelerador de roteamento personalizado

O `add-custom-routing-endpoints` exemplo a seguir adiciona um endpoint de VPC sub-rede a um grupo de endpoints para um acelerador de roteamento personalizado.

```
aws globalaccelerator add-custom-routing-endpoints \
  --endpoint-group-
  arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
  abcd-1234abcdefg/Listener/0123vxyz/endpoint-group/4321abcd \
  --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

Saída:

```
{
  "EndpointDescriptions": [
    {
      "EndpointId": "subnet-1234567890abcdef0"
    }
  ],
  "EndpointGroupArn": "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefg/Listener/0123vxyz/endpoint-group/4321abcd"
}
```

Para obter mais informações, consulte [endpoints de VPC sub-rede para aceleradores de roteamento personalizados no Global Accelerator no AWS Global Accelerator Developer Guide](#).AWS

- Para API obter detalhes, consulte [AddCustomRoutingEndpoints](#) na Referência de AWS CLI Comandos.

advertise-byoip-cidr

O código de exemplo a seguir mostra como usar `advertise-byoip-cidr`.

AWS CLI

Para anunciar um intervalo de endereços

O `advertise-byoip-cidr` exemplo a seguir solicita AWS a publicidade de um intervalo de endereços que você provisionou para uso com seus recursos. AWS

```
aws globalaccelerator advertise-byoip-cidr \  
  --cidr 198.51.100.0/24
```

Saída:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_ADVERTISING"  
  }  
}
```

Para obter mais informações, consulte [Traga seu próprio endereço IP no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [AdvertiseByoipCidr](#) na Referência de AWS CLI Comandos.

allow-custom-routing-traffic

O código de exemplo a seguir mostra como usar `allow-custom-routing-traffic`.

AWS CLI

Para permitir o tráfego para destinos específicos de EC2 instâncias da Amazon em uma VPC sub-rede para um acelerador de roteamento personalizado

O `allow-custom-routing-traffic` exemplo a seguir especifica que o tráfego é permitido para determinados endereços IP da EC2 instância (destino) da Amazon e portas para que um endpoint de VPC sub-rede em um acelerador de roteamento personalizado possa receber tráfego.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [endpoints de VPC sub-rede para aceleradores de roteamento personalizados no Global Accelerator no AWS Global Accelerator Developer Guide](#).AWS

- Para API obter detalhes, consulte [AllowCustomRoutingTraffic](#) na Referência de AWS CLI Comandos.

create-accelerator

O código de exemplo a seguir mostra como usar create-accelerator.

AWS CLI

Para criar um acelerador

O create-accelerator exemplo a seguir cria um acelerador com duas tags com dois endereços IP BYOIP estáticos. Você deve especificar a US-West-2 (Oregon) região para criar ou atualizar um acelerador.

```
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
```

Saída:

```
{  
  "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
  }
}

```

Para obter mais informações, consulte [Aceleradores no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [CreateAccelerator](#) na Referência de AWS CLI Comandos.

create-custom-routing-accelerator

O código de exemplo a seguir mostra como usar `create-custom-routing-accelerator`.

AWS CLI

Para criar um acelerador de roteamento personalizado

O `create-custom-routing-accelerator` exemplo a seguir cria um acelerador de roteamento personalizado com as tags 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

```

Saída:

```

{
  "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
    }
  }
}

```

Para obter mais informações, consulte [Aceleradores de roteamento personalizados no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [CreateCustomRoutingAccelerator](#) na Referência de AWS CLI Comandos.

create-custom-routing-endpoint-group

O código de exemplo a seguir mostra como usar `create-custom-routing-endpoint-group`.

AWS CLI

Para criar um grupo de endpoints para um acelerador de roteamento personalizado

O `create-custom-routing-endpoint-group` exemplo a seguir cria um grupo de endpoints para um acelerador de roteamento personalizado.

```

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"
```

Saída:

```
{
  "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": []
  }
}
```

Para obter mais informações, consulte [Grupos de endpoints para aceleradores de roteamento personalizados no Global Accelerator no AWS Global Accelerator Developer Guide AWS](#) .

- Para API obter detalhes, consulte [CreateCustomRoutingEndpointGroup](#) na Referência de AWS CLI Comandos.

create-custom-routing-listener

O código de exemplo a seguir mostra como usar `create-custom-routing-listener`.

AWS CLI

Para criar um ouvinte para um acelerador de roteamento personalizado

O `create-custom-routing-listener` exemplo a seguir cria um ouvinte com um intervalo de portas de 5000 a 10000 para um acelerador de roteamento personalizado.

```
aws globalaccelerator create-custom-routing-listener \
```

```
--accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
--port-ranges FromPort=5000,ToPort=10000
```

Saída:

```
{
  "Listener": {
    "PortRange": [
      "FromPort": 5000,
      "ToPort": 10000
    ],
    "ListenerArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz"
  }
}
```

Para obter mais informações, consulte [Ouvintes de aceleradores de roteamento personalizados no Global Accelerator no AWS Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [CreateCustomRoutingListener](#) na Referência de AWS CLI Comandos.

create-endpoint-group

O código de exemplo a seguir mostra como usar `create-endpoint-group`.

AWS CLI

Para criar um grupo de endpoints

O `create-endpoint-group` exemplo a seguir cria um grupo de endpoints com um endpoint.

```
aws globalaccelerator create-endpoint-group \
--listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
--endpoint-group-region us-east-1 \
--endpoint-configurations EndpointId=i-1234567890abcdef0,Weight=128
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Grupos de endpoints no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [CreateEndpointGroup](#) na Referência de AWS CLI Comandos.

create-listener

O código de exemplo a seguir mostra como usar `create-listener`.

AWS CLI

Para criar um ouvinte

O `create-listener` exemplo a seguir cria um ouvinte com duas portas.

```
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
```

Saída:

```
{
  "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"
    }
}

```

Para obter mais informações, consulte [Listeners in AWS Global Accelerator no Global Accelerator AWS Developer Guide](#).

- Para API obter detalhes, consulte [CreateListener](#) na Referência de AWS CLI Comandos.

deny-custom-routing-traffic

O código de exemplo a seguir mostra como usar deny-custom-routing-traffic.

AWS CLI

Para especificar um endereço de destino que não pode receber tráfego em um acelerador de roteamento personalizado

O deny-custom-routing-traffic exemplo a seguir especifica o endereço ou endereços de destino em um endpoint de sub-rede que não pode receber tráfego para um acelerador de roteamento personalizado. Para especificar mais de um endereço de destino, separe os endereços com um espaço. Não há resposta para uma deny-custom-routing-traffic chamada bem-sucedida.

```

aws globalaccelerator deny-custom-routing-traffic \
  --endpoint-group-
  arn "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab8888example" \
  --endpoint-id "subnet-abcd123example" \

```

```
--destination-addresses "198.51.100.52"
```

Este comando não produz saída.

Para obter mais informações, consulte [endpoints de VPC sub-rede para aceleradores de roteamento personalizados no Global Accelerator no AWS Global Accelerator Developer Guide.AWS](#)

- Para API obter detalhes, consulte [DenyCustomRoutingTraffic](#) na Referência de AWS CLI Comandos.

deprovision-byoip-cidr

O código de exemplo a seguir mostra como usar `deprovision-byoip-cidr`.

AWS CLI

Para desprovisionar um intervalo de endereços

O `deprovision-byoip-cidr` exemplo a seguir libera o intervalo de endereços especificado que você provisionou para usar com seus AWS recursos.

```
aws globalaccelerator deprovision-byoip-cidr \  
  --cidr "198.51.100.0/24"
```

Saída:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_DEPROVISIONING"  
  }  
}
```

Para obter mais informações, consulte [Traga seu próprio endereço IP no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [DeprovisionByoipCidr](#) na Referência de AWS CLI Comandos.

describe-accelerator-attributes

O código de exemplo a seguir mostra como usar `describe-accelerator-attributes`.

AWS CLI

Para descrever os atributos de um acelerador

O `describe-accelerator-attributes` exemplo a seguir recupera os detalhes do atributo de um acelerador.

```
aws globalaccelerator describe-accelerator-attributes \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

Saída:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": true  
    "FlowLogsS3Bucket": flowlogs-abc  
    "FlowLogsS3Prefix": bucketprefix-abc  
  }  
}
```

Para obter mais informações, consulte [Aceleradores no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [DescribeAcceleratorAttributes](#) na Referência de AWS CLI Comandos.

`describe-accelerator`

O código de exemplo a seguir mostra como usar `describe-accelerator`.

AWS CLI

Para descrever um acelerador

O `describe-accelerator` exemplo a seguir recupera os detalhes sobre o acelerador especificado.

```
aws globalaccelerator describe-accelerator \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```


Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Aceleradores no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [DescribeAccelerator](#) na Referência de AWS CLI Comandos.

describe-custom-routing-accelerator-attributes

O código de exemplo a seguir mostra como usar `describe-custom-routing-accelerator-attributes`.

AWS CLI

Para descrever os atributos de um acelerador de roteamento personalizado

O `describe-custom-routing-accelerator-attributes` exemplo a seguir descreve os atributos de um acelerador de roteamento personalizado.

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Saída:

```
{
  "AcceleratorAttributes": {
    "FlowLogsEnabled": false
  }
}
```

Para obter mais informações, consulte [Aceleradores de roteamento personalizados no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [DescribeCustomRoutingAcceleratorAttributes](#) na Referência de AWS CLI Comandos.

describe-custom-routing-accelerator

O código de exemplo a seguir mostra como usar `describe-custom-routing-accelerator`.

AWS CLI

Para descrever um acelerador de roteamento personalizado

O `describe-custom-routing-accelerator` exemplo a seguir recupera os detalhes sobre o acelerador de roteamento personalizado especificado.

```
aws globalaccelerator describe-custom-routing-accelerator \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Saída:

```
{
  "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
  }
}

```

Para obter mais informações, consulte [Aceleradores de roteamento personalizados no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [DescribeCustomRoutingAccelerator](#) na Referência de AWS CLI Comandos.

describe-custom-routing-endpoint-group

O código de exemplo a seguir mostra como usar `describe-custom-routing-endpoint-group`.

AWS CLI

Para descrever um grupo de endpoints para um acelerador de roteamento personalizado

O `describe-custom-routing-endpoint-group` exemplo a seguir descreve um grupo de endpoints para um acelerador de roteamento personalizado.

```

aws globalaccelerator describe-custom-routing-endpoint-group \
  --endpoint-group-
  arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
  abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab8888example

```

Saída:

```

{
  "EndpointGroup": {

```

```

    "EndpointGroupArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab8888example",
    "EndpointGroupRegion": "us-east-2",
    "DestinationDescriptions": [
      {
        "FromPort": 5000,
        "ToPort": 10000,
        "Protocols": [
          "UDP"
        ]
      }
    ],
    "EndpointDescriptions": [
      {
        "EndpointId": "subnet-1234567890abcdef0"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Grupos de endpoints para aceleradores de roteamento personalizados no Global Accelerator no AWS Global Accelerator Developer Guide AWS](#) .

- Para API obter detalhes, consulte [DescribeCustomRoutingEndpointGroup](#) na Referência de AWS CLI Comandos.

describe-custom-routing-listener

O código de exemplo a seguir mostra como usar `describe-custom-routing-listener`.

AWS CLI

Para descrever um ouvinte de um acelerador de roteamento personalizado

O `describe-custom-routing-listener` exemplo a seguir descreve um ouvinte para um acelerador de roteamento personalizado.

```

aws globalaccelerator describe-custom-routing-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234

```

Saída:

```
{
  "Listener": {
    "PortRanges": [
      "FromPort": 5000,
      "ToPort": 10000
    ],
    "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234"
  }
}
```

Para obter mais informações, consulte [Ouvintes de aceleradores de roteamento personalizados no Global Accelerator no AWS Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [DescribeCustomRoutingListener](#) na Referência de AWS CLI Comandos.

describe-endpoint-group

O código de exemplo a seguir mostra como usar `describe-endpoint-group`.

AWS CLI

Para descrever um grupo de endpoints

O `describe-endpoint-group` exemplo a seguir recupera detalhes sobre um grupo de endpoints com os seguintes endpoints: uma EC2 instância da AmazonALB, uma e uma. 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
```

Saída:

```
{
  "EndpointGroup": {
    "TrafficDialPercentage": 100.0,
    "EndpointDescriptions": [
      {
```

```

        "Weight": 128,
        "EndpointId": "i-1234567890abcdef0"
    },
    {
        "Weight": 128,
        "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/app/ALBTesting/alb01234567890xyz"
    },
    {
        "Weight": 128,
        "EndpointId": "arn:aws:elasticloadbalancing:us-
east-1:000123456789:loadbalancer/net/NLBTesting/alb01234567890qrs"
    }
],
"EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
"EndpointGroupRegion": "us-east-1"
}
}

```

Para obter mais informações, consulte [Grupos de endpoints no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [DescribeEndpointGroup](#) na Referência de AWS CLI Comandos.

describe-listener

O código de exemplo a seguir mostra como usar `describe-listener`.

AWS CLI

Para descrever um ouvinte

O `describe-listener` exemplo a seguir descreve um ouvinte.

```

aws globalaccelerator describe-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234

```

Saída:

```
{
  "Listener": {
    "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
    "PortRanges": [
      {
        "FromPort": 80,
        "ToPort": 80
      }
    ],
    "Protocol": "TCP",
    "ClientAffinity": "NONE"
  }
}
```

Para obter mais informações, consulte [Listeners in AWS Global Accelerator no Global Accelerator AWS Developer Guide](#).

- Para API obter detalhes, consulte [DescribeListener](#) na Referência de AWS CLI Comandos.

list-accelerators

O código de exemplo a seguir mostra como usar `list-accelerators`.

AWS CLI

Para listar seus aceleradores

O `list-accelerators` exemplo a seguir lista os aceleradores em sua AWS conta. Essa conta tem dois aceleradores.

```
aws globalaccelerator list-accelerators
```

Saída:

```
{
  "Accelerators": [
    {
      "AcceleratorArn":
"arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
```


- Para API obter detalhes, consulte [ListAccelerators](#) na Referência de AWS CLI Comandos.

list-byoip-cidr

O código de exemplo a seguir mostra como usar `list-byoip-cidr`.

AWS CLI

Para listar seus intervalos de endereços

O `list-byoip-cidr` exemplo a seguir lista os intervalos de endereços “traga seu próprio endereço IP” (BYOIP) que você provisionou para uso com o Global Accelerator.

```
aws globalaccelerator list-byoip-cidrs
```

Saída:

```
{
  "ByoipCidrs": [
    {
      "Cidr": "198.51.100.0/24",
      "State": "READY"
    }
    {
      "Cidr": "203.0.113.25/24",
      "State": "READY"
    }
  ]
}
```

Para obter mais informações, consulte [Traga seu próprio endereço IP no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [ListByoipCidrs](#) na Referência de AWS CLI Comandos.

list-custom-routing-accelerators

O código de exemplo a seguir mostra como usar `list-custom-routing-accelerators`.

AWS CLI

Para listar seus aceleradores de roteamento personalizados


```

    {
      "FromPort": 80,
      "ToPort": 80,
      "Protocols": [
        "TCP",
        "UDP"
      ]
    }
  ]
  "EndpointDescriptions": [
    {
      "EndpointId": "subnet-abcd123example"
    }
  ]
}
]
}

```

Para obter mais informações, consulte [Grupos de endpoints para aceleradores de roteamento personalizados no Global Accelerator no AWS Global Accelerator Developer Guide](#) AWS .

- Para API obter detalhes, consulte [ListCustomRoutingEndpointGroups](#) na Referência de AWS CLI Comandos.

list-custom-routing-listeners

O código de exemplo a seguir mostra como usar `list-custom-routing-listeners`.

AWS CLI

Para listar ouvintes de aceleradores de roteamento personalizados

O `list-custom-routing-listeners` exemplo a seguir lista os ouvintes de um acelerador de roteamento personalizado.

```

aws globalaccelerator list-custom-routing-listeners \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Saída:

```

{
  "Listeners": [

```

```
{
  "ListenerArn":
  "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
  "PortRanges": [
    {
      "FromPort": 5000,
      "ToPort": 10000
    }
  ],
  "Protocol": "TCP"
}
]
```

Para obter mais informações, consulte [Ouvintes de aceleradores de roteamento personalizados no Global Accelerator no AWS Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [ListCustomRoutingListeners](#) na Referência de AWS CLI Comandos.

list-custom-routing-port-mappings-by-destination

O código de exemplo a seguir mostra como usar `list-custom-routing-port-mappings-by-destination`.

AWS CLI

Para listar os mapeamentos de portas para um destino específico do acelerador de roteamento personalizado

O `list-custom-routing-port-mappings-by-destination` exemplo a seguir fornece os mapeamentos de portas para um EC2 servidor de destino específico (no endereço de destino) para um acelerador de roteamento personalizado.

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \
  --endpoint-id subnet-abcd123example \
  --destination-address 198.51.100.52
```

Saída:

```
{
```

```

    "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Como os aceleradores de roteamento personalizados funcionam no AWS Global Accelerator no Global Accelerator Developer AWS Guide](#).

- Para API obter detalhes, consulte [ListCustomRoutingPortMappingsByDestination](#) na Referência de AWS CLI Comandos.

list-custom-routing-port-mappings

O código de exemplo a seguir mostra como usar `list-custom-routing-port-mappings`.

AWS CLI

Para listar os mapeamentos de portas em um acelerador de roteamento personalizado

O `list-custom-routing-port-mappings` exemplo a seguir fornece uma lista parcial dos mapeamentos de portas em um acelerador de roteamento personalizado.

```
aws globalaccelerator list-custom-routing-port-mappings \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

Para obter mais informações, consulte [Como os aceleradores de roteamento personalizados funcionam no AWS Global Accelerator no Global Accelerator Developer AWS Guide](#).

- Para API obter detalhes, consulte [ListCustomRoutingPortMappings](#) na Referência de AWS CLI Comandos.

list-endpoint-groups

O código de exemplo a seguir mostra como usar `list-endpoint-groups`.

AWS CLI

Para listar grupos de endpoints

O `list-endpoint-groups` exemplo a seguir lista os grupos de endpoints de um ouvinte. Esse ouvinte tem dois grupos de endpoints.

```
aws globalaccelerator --region us-west-2 list-endpoint-groups \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Saída:

```
{
  "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
  }
]
}

```

Para obter mais informações, consulte [Endpoint Groups no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [ListEndpointGroups](#) na Referência de AWS CLI Comandos.

list-listeners

O código de exemplo a seguir mostra como usar `list-listeners`.

AWS CLI

Para listar ouvintes

O `list-listeners` exemplo a seguir lista os ouvintes de um acelerador.

```

aws globalaccelerator list-listeners \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Saída:

```

{
  "Listeners": [
    {
      "ListenerArn":
        "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",
      "PortRanges": [
        {

```

```

        "FromPort": 80,
        "ToPort": 80
      }
    ],
    "Protocol": "TCP",
    "ClientAffinity": "NONE"
  }
]
}

```

Para obter mais informações, consulte [Listeners in AWS Global Accelerator no Global Accelerator AWS Developer Guide](#).

- Para API obter detalhes, consulte [ListListeners](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um acelerador

O `list-tags-for-resource` exemplo a seguir lista as tags de um acelerador específico.

```

aws globalaccelerator list-tags-for-resource \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Saída:

```

{
  "Tags": [
    {
      "Key": "Project",
      "Value": "A123456"
    }
  ]
}

```

Para obter mais informações, consulte [Marcação no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

provision-byoip-cidr

O código de exemplo a seguir mostra como usar `provision-byoip-cidr`.

AWS CLI

Para provisionar um intervalo de endereços

O `provision-byoip-cidr` exemplo a seguir provisiona o intervalo de endereços especificado para uso com seus AWS recursos.

```
aws globalaccelerator provision-byoip-cidr \  
  --cidr 192.0.2.250/24 \  
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Saída:

```
{  
  "ByoipCidr": {  
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_PROVISIONING"  
  }  
}
```

Para obter mais informações, consulte [Traga seu próprio endereço IP no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [ProvisionByoipCidr](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um acelerador

O `tag-resource` exemplo a seguir adiciona as tags `Name` e `Project` a um acelerador, junto com os valores correspondentes para cada uma.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover uma tag de um acelerador

O untag-resource exemplo a seguir remove as tags Nome e Projeto de um acelerador.

```
aws globalaccelerator untag-resource \  
  --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --tag-keys Key="Name" Key="Project"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-accelerator-attributes

O código de exemplo a seguir mostra como usar update-accelerator-attributes.

AWS CLI

Para atualizar os atributos de um acelerador

O `update-accelerator-attributes` exemplo a seguir atualiza um acelerador para ativar os registros de fluxo. Você deve especificar a US-West-2 (Oregon) região para criar ou atualizar os atributos do acelerador.

```
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
```

Saída:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": true  
    "FlowLogsS3Bucket": flowlogs-abc  
    "FlowLogsS3Prefix": bucketprefix-abc  
  }  
}
```

Para obter mais informações, consulte [Aceleradores no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [UpdateAcceleratorAttributes](#) na Referência de AWS CLI Comandos.

update-accelerator

O código de exemplo a seguir mostra como usar `update-accelerator`.

AWS CLI

Para atualizar um acelerador

O `update-accelerator` exemplo a seguir modifica um acelerador para alterar o nome do acelerador para `ExampleAcceleratorNew`. Você deve especificar a US-West-2 (Oregon) região para criar ou atualizar aceleradores.

```
aws globalaccelerator update-accelerator \  
  --accelerator-name ExampleAcceleratorNew
```

```
--accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
--name ExampleAcceleratorNew
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Aceleradores no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [UpdateAccelerator](#) na Referência de AWS CLI Comandos.

update-custom-routing-accelerator-attributes

O código de exemplo a seguir mostra como usar `update-custom-routing-accelerator-attributes`.

AWS CLI

Para atualizar os atributos de um acelerador de roteamento personalizado

O `update-custom-routing-accelerator-attributes` exemplo a seguir atualiza um acelerador de roteamento personalizado para habilitar registros de fluxo.

```
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
```

Saída:

```
{
  "AcceleratorAttributes": {
    "FlowLogsEnabled": true
    "FlowLogsS3Bucket": flowlogs-abc
    "FlowLogsS3Prefix": bucketprefix-abc
  }
}
```

Para obter mais informações, consulte [Aceleradores de roteamento personalizados no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [UpdateCustomRoutingAcceleratorAttributes](#) na Referência de AWS CLI Comandos.

update-custom-routing-accelerator

O código de exemplo a seguir mostra como usar `update-custom-routing-accelerator`.

AWS CLI

Para atualizar um acelerador de roteamento personalizado

O `update-custom-routing-accelerator` exemplo a seguir modifica um acelerador de roteamento personalizado para alterar o nome do acelerador.

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
  --name ExampleCustomRoutingAcceleratorNew
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Aceleradores de roteamento personalizados no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [UpdateCustomRoutingAccelerator](#) na Referência de AWS CLI Comandos.

update-custom-routing-listener

O código de exemplo a seguir mostra como usar `update-custom-routing-listener`.

AWS CLI

Para atualizar um ouvinte para um acelerador de roteamento personalizado

O `update-custom-routing-listener` exemplo a seguir atualiza um ouvinte para alterar o intervalo de portas.


```
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
```

Saída:

```
{
  "Listener": {
    "ListenerArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz
    "PortRanges": [
      {
        "FromPort": 10000,
        "ToPort": 20000
      }
    ],
    "Protocol": "TCP"
  }
}
```

Para obter mais informações, consulte [Ouvintes de aceleradores de roteamento personalizados no Global Accelerator no AWS Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [UpdateCustomRoutingListener](#) na Referência de AWS CLI Comandos.

update-endpoint-group

O código de exemplo a seguir mostra como usar update-endpoint-group.

AWS CLI

Para atualizar um grupo de endpoints

O update-endpoint-group exemplo a seguir adiciona três endpoints a um grupo de endpoints: um endereço IP elásticoALB, um e um. NLB

```
aws globalaccelerator update-endpoint-group \
  --endpoint-group-
arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
```

```

abcd-1234abcdefgh/Listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab8888example \
  --endpoint-configurations \
    EndpointId=eipalloc-eip01234567890abc,Weight=128 \
    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

```

Saída:

```

{
  "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-1234abcdefgh/Listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefgh",
    "EndpointGroupRegion": "us-east-1"
  }
}

```

Para obter mais informações, consulte [Grupos de endpoints no AWS Global Accelerator no Guia do desenvolvedor do AWS Global Accelerator](#).

- Para API obter detalhes, consulte [UpdateEndpointGroup](#) na Referência de AWS CLI Comandos.

update-listener

O código de exemplo a seguir mostra como usar `update-listener`.

AWS CLI

Para atualizar um ouvinte

O `update-listener` exemplo a seguir atualiza um ouvinte para alterar a porta para 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
```

Saída:

```
{  
  "Listener": {  
    "ListenerArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz"  
    "PortRanges": [  
      {  
        "FromPort": 100,  
        "ToPort": 100  
      }  
    ],  
    "Protocol": "TCP",  
    "ClientAffinity": "NONE"  
  }  
}
```

Para obter mais informações, consulte [Listeners in AWS Global Accelerator no Global Accelerator AWS Developer Guide](#).

- Para API obter detalhes, consulte [UpdateListener](#) na Referência de AWS CLI Comandos.

withdraw-byoip-cidr

O código de exemplo a seguir mostra como usar `withdraw-byoip-cidr`.

AWS CLI

Para retirar um intervalo de endereços

O `withdraw-byoip-cidr` exemplo a seguir retira um intervalo de endereços do AWS Global Accelerator que você anunciou anteriormente para uso com seus recursos. AWS

```
aws globalaccelerator withdraw-byoip-cidr \  
  --cidr 192.0.2.250/24
```

Saída:

```
{  
  "ByoipCidr": {  
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_WITHDRAWING"  
  }  
}
```

Para obter mais informações, consulte [Traga seu próprio endereço IP no AWS Global Accelerator no Guia](#) do desenvolvedor do AWS Global Accelerator.

- Para API obter detalhes, consulte [WithdrawByoipCidr](#) na Referência de AWS CLI Comandos.

AWS Glue exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Glue.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-stop-job-run

O código de exemplo a seguir mostra como usar batch-stop-job-run.

AWS CLI

Para interromper a execução do trabalho

O batch-stop-job-run exemplo a seguir interrompe a execução de um trabalho.

```
aws glue batch-stop-job-run \  
  --job-name "my-testing-job" \  
  --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Execuções de trabalhos](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [BatchStopJobRun](#) na Referência de AWS CLI Comandos.

create-connection

O código de exemplo a seguir mostra como usar `create-connection`.

AWS CLI

Para criar uma conexão para os armazenamentos de dados do AWS Glue

O `create-connection` exemplo a seguir cria uma conexão no AWS Glue Data Catalog que fornece informações de conexão para um armazenamento de dados 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo conexões no catálogo de dados do AWS Glue](#) no AWS Glue Developer Guide.

- Para API obter detalhes, consulte [CreateConnection](#) na Referência de AWS CLI Comandos.

create-database

O código de exemplo a seguir mostra como usar `create-database`.

AWS CLI

Para criar um banco de dados

O `create-database` exemplo a seguir cria um banco de dados no AWS Glue Data Catalog.

```
aws glue create-database \  
  --database-input "{\"Name\":\"tempdb\"}" \  
  --profile my_profile \  
  --endpoint https://glue.us-east-1.amazonaws.com
```

Este comando não produz saída.

Para obter mais informações, consulte [Definir um banco de dados no seu Catálogo de Dados](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [CreateDatabase](#) na Referência de AWS CLI Comandos.

`create-job`

O código de exemplo a seguir mostra como usar `create-job`.

AWS CLI

Para criar um trabalho para transformar dados

O exemplo de `create-job` a seguir cria um trabalho de streaming que executa um script armazenado no 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
```

Conteúdo de `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()
  }
}

```

Saída:

```

{
  "Name": "my-testing-job"
}

```

Para obter mais informações, consulte [Authoring Jobs in AWS Glue no Glue Developer Guide](#).AWS

- Para API obter detalhes, consulte [CreateJob](#) na Referência de AWS CLI Comandos.

create-table

O código de exemplo a seguir mostra como usar create-table.

AWS CLI

Exemplo 1: Para criar uma tabela para um stream de dados do Kinesis

O `create-table` exemplo a seguir cria uma tabela no AWS Glue Data Catalog que descreve um stream de dados do 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo tabelas no catálogo de dados do AWS Glue](#) no AWS Glue Developer Guide.

Exemplo 2: Para criar uma tabela para um armazenamento de dados do Kafka

O `create-table` exemplo a seguir cria uma tabela no AWS Glue Data Catalog que descreve um armazenamento de dados 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

```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo tabelas no catálogo de dados do AWS Glue](#) no AWS Glue Developer Guide.

Exemplo 3: Para criar uma tabela para um armazenamento de dados AWS do S3

O `create-table` exemplo a seguir cria uma tabela no AWS Glue Data Catalog que descreve um armazenamento de dados do 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo tabelas no catálogo de dados do AWS Glue](#) no AWS Glue Developer Guide.

- Para API obter detalhes, consulte [CreateTable](#) na Referência de AWS CLI Comandos.

delete-job

O código de exemplo a seguir mostra como usar delete-job.

AWS CLI

Para excluir um trabalho

O exemplo de delete-job a seguir exclui um trabalho que não é mais necessário.

```
aws glue delete-job \  
--job-name my-testing-job
```

Saída:

```
{  
  "JobName": "my-testing-job"  
}
```

Para obter mais informações, consulte [Trabalhando com trabalhos no AWS Glue Console](#) no AWS Glue Developer Guide.

- Para API obter detalhes, consulte [DeleteJob](#) na Referência de AWS CLI Comandos.

get-databases

O código de exemplo a seguir mostra como usar get-databases.

AWS CLI

Para listar as definições de alguns ou de todos os bancos de dados no AWS Glue Data Catalog

O exemplo de `get-databases` a seguir retorna informações sobre os bancos de dados no Catálogo de Dados.

```
aws glue get-databases
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Definir um banco de dados no seu Catálogo de Dados](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [GetDatabases](#) na Referência de AWS CLI Comandos.

get-job-run

O código de exemplo a seguir mostra como usar `get-job-run`.

AWS CLI

Para obter informações sobre uma execução de trabalho

O exemplo `get-job-run` a seguir recupera informações sobre uma execução de trabalho.

```
aws glue get-job-run \  
  --job-name "Combine legislators data" \  
  --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Execuções de trabalhos](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [GetJobRun](#) na Referência de AWS CLI Comandos.

get-job-runs

O código de exemplo a seguir mostra como usar `get-job-runs`.

AWS CLI

Para obter informações sobre todas as execuções de um trabalho

O exemplo `get-job-runs` a seguir recupera informações sobre execuções de trabalho para um trabalho.

```
aws glue get-job-runs \  
  --job-name "my-testing-job"
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Execuções de trabalhos](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [GetJobRuns](#) na Referência de AWS CLI Comandos.

get-job

O código de exemplo a seguir mostra como usar `get-job`.

AWS CLI

Para recuperar informações sobre um trabalho

O exemplo de `get-job` a seguir recupera informações sobre um trabalho.

```

aws glue get-job \
  --job-name my-testing-job

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Trabalhos](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [GetJob](#) na Referência de AWS CLI Comandos.

get-plan

O código de exemplo a seguir mostra como usar `get-plan`.

AWS CLI

Para obter o código gerado para mapear dados das tabelas de origem para as tabelas de destino

O código a seguir é `get-plan` recuperado para mapear colunas da fonte de dados para o destino de dados.

```

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"

```

Saída:

```

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()
}
}
```

Para obter mais informações, consulte [Editando scripts no AWS Glue](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [GetPlan](#) na Referência de AWS CLI Comandos.

get-tables

O código de exemplo a seguir mostra como usar `get-tables`.

AWS CLI

Para listar as definições de algumas ou de todas as tabelas no banco de dados especificado

O exemplo de `get-tables` a seguir retorna informações sobre as tabelas no banco de dados especificado.

```
aws glue get-tables --database-name 'tempdb'
```

Saída:

```
{
  "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"
}
]
}
```

Para obter mais informações, consulte [Definindo tabelas no catálogo de dados do AWS Glue](#) no AWS Glue Developer Guide.

- Para API obter detalhes, consulte [GetTables](#) na Referência de AWS CLI Comandos.

start-crawler

O código de exemplo a seguir mostra como usar `start-crawler`.

AWS CLI

Para iniciar um crawler

O exemplo de `start-crawler` a seguir inicia um crawler.

```
aws glue start-crawler --name my-crawler
```

Saída:

```
None
```

Para obter mais informações, consulte [Definição de crawlers](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [StartCrawler](#) na Referência de AWS CLI Comandos.

start-job-run

O código de exemplo a seguir mostra como usar `start-job-run`.

AWS CLI

Para iniciar a execução de um trabalho

O exemplo de `start-job-run` a seguir inicia um trabalho.

```
aws glue start-job-run \  
  --job-name my-job
```

Saída:

```
{  
  "JobRunId":  
  "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"  
}
```

Para obter mais informações, consulte [Criação de trabalhos](#) no Guia do desenvolvedor do AWS Glue.

- Para API obter detalhes, consulte [StartJobRun](#) na Referência de AWS CLI Comandos.

GuardDuty exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with GuardDuty.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-invitation

O código de exemplo a seguir mostra como usar `accept-invitation`.

AWS CLI

Para aceitar um convite para se tornar uma conta GuardDuty membro na região atual

O `accept-invitation` exemplo a seguir mostra como aceitar um convite para se tornar uma conta GuardDuty membro na região atual.

```
aws guardduty accept-invitation \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --master-id 123456789111 \  
  --invitation-id d6b94fb03a66ff665f7db8764example
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de GuardDuty contas por convite](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [AcceptInvitation](#) na Referência de AWS CLI Comandos.

archive-findings

O código de exemplo a seguir mostra como usar `archive-findings`.

AWS CLI

Para arquivar descobertas na região atual

Este exemplo mostra como arquivar descobertas na região atual.

```
aws guardduty archive-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de GuardDuty contas por convite](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [ArchiveFindings](#) na Referência de AWS CLI Comandos.

create-detector

O código de exemplo a seguir mostra como usar `create-detector`.

AWS CLI

Para habilitar GuardDuty na região atual

Este exemplo mostra como criar um novo detector, que habilita GuardDuty, na região atual. :

```
aws guardduty create-detector \  
  --enable
```

Saída:

```
{  
  "DetectorId": "b6b992d6d2f48e64bc59180bfexample"  
}
```

Para obter mais informações, consulte [Habilitar a Amazon GuardDuty](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [CreateDetector](#) na Referência de AWS CLI Comandos.

create-filter

O código de exemplo a seguir mostra como usar `create-filter`.

AWS CLI

Para criar um novo filtro para a região atual

Este exemplo cria um filtro que corresponde a todas as descobertas do portscan, por exemplo, criadas a partir de uma imagem específica. :

```
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"]}}}'
```

Saída:

```
{  
  "Name": "myFilter"  
}
```

Para obter mais informações, consulte [Filtragem de descobertas](#) no Guia do GuardDuty usuário.

- Para API obter detalhes, consulte [CreateFilter](#) na Referência de AWS CLI Comandos.

create-ip-set

O código de exemplo a seguir mostra como usar create-ip-set.

AWS CLI

Para criar um conjunto de IP confiável

O create-ip-set exemplo a seguir cria e ativa um conjunto de IP confiável na região atual.

```
aws guardduty create-ip-set \  
--detector-id 12abc34d567e8fa901bc2d34eexample \  
--name new-ip-set \  
--format TXT \  
--location s3://AWSDOC-EXAMPLE-BUCKET/customtrustlist.csv \  
--activate
```

Saída:

```
{  
  "IpSetId": "d4b94fc952d6912b8f3060768example"  
}
```

Para obter mais informações, consulte [Trabalhando com listas de IPs confiáveis e listas de ameaças](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [CreatepSet](#) na Referência de AWS CLI Comandos.

create-members

O código de exemplo a seguir mostra como usar `create-members`.

AWS CLI

Para associar um novo membro à sua conta GuardDuty principal na região atual.

Este exemplo mostra como associar contas de membros a serem gerenciadas pela conta atual como GuardDuty mestre.

```
aws guardduty create-members
  --detector-id b6b992d6d2f48e64bc59180bfexample \
  --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

Saída:

```
{
  "UnprocessedAccounts": []
}
```

Para obter mais informações, consulte [Gerenciamento de várias contas](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [CreateMembers](#) na Referência de AWS CLI Comandos.

create-publishing-destination

O código de exemplo a seguir mostra como usar `create-publishing-destination`.

AWS CLI

Para criar um destino de publicação para o GuardDuty qual exportar descobertas na região atual.

Este exemplo mostra como criar um destino de publicação para GuardDuty descobertas.

```
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
```

Saída:

```
{  
  "DestinationId": "46b99823849e1bbc242dfbe3cexample"  
}
```

Para obter mais informações, consulte [Exportação de descobertas](#) no Guia do GuardDuty usuário.

- Para API obter detalhes, consulte [CreatePublishingDestination](#) na Referência de AWS CLI Comandos.

create-sample-findings

O código de exemplo a seguir mostra como usar `create-sample-findings`.

AWS CLI

Para criar amostras de GuardDuty resultados na região atual.

Este exemplo mostra como criar uma descoberta de amostra dos tipos fornecidos.

```
aws guardduty create-sample-findings \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

Este comando não produz saída.

Para obter mais informações, consulte [Exemplos de descobertas](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [CreateSampleFindings](#) na Referência de AWS CLI Comandos.

create-threat-intel-set

O código de exemplo a seguir mostra como usar `create-threat-intel-set`.

AWS CLI

Para criar um novo conjunto de informações sobre ameaças na região atual.

Este exemplo mostra como fazer o upload de um conjunto de informações sobre ameaças GuardDuty e ativá-lo imediatamente.

```
aws guardduty create-threat-intel-set \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --name myThreatSet \  
  --format TXT \  
  --location s3://EXAMPLEBUCKET/threatlist.csv \  
  --activate
```

Saída:

```
{  
  "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"  
}
```

Para obter mais informações, consulte [IPs confiáveis e listas de ameaças](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [CreateThreatIntelSet](#) na Referência de AWS CLI Comandos.

decline-invitations

O código de exemplo a seguir mostra como usar `decline-invitations`.

AWS CLI

Para recusar um convite para que o Guardduty seja gerenciado por outra conta na região atual.

Este exemplo mostra como recusar um convite de associação.

```
aws guardduty decline-invitations \  
  --account-ids 111122223333
```

Saída:

```
{  
  "UnprocessedAccounts": []  
}
```



```
}
```

Para obter mais informações, consulte [Gerenciamento de GuardDuty contas por convite](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [DeclineInvitations](#) na Referência de AWS CLI Comandos.

delete-detector

O código de exemplo a seguir mostra como usar `delete-detector`.

AWS CLI

Para excluir e desativar GuardDuty um detector na região atual.

Este exemplo mostra como excluir um detector. Se for bem-sucedido, isso será desativado GuardDuty na região associada a esse detector.

```
aws guardduty delete-detector \  
  --detector-id b6b992d6d2f48e64bc59180bfexample
```

Este comando não produz saída.

Para obter mais informações, consulte [Suspensão ou desativação GuardDuty](#) no Guia do GuardDuty usuário.

- Para API obter detalhes, consulte [DeleteDetector](#) na Referência de AWS CLI Comandos.

delete-filter

O código de exemplo a seguir mostra como usar `delete-filter`.

AWS CLI

Para excluir um filtro existente na região atual

Este exemplo mostra como criar um filtro para excluir.

```
aws guardduty delete-filter \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --filter-name byebyeFilter
```

Este comando não produz saída.

Para obter mais informações, consulte [Filtragem de descobertas](#) no Guia do GuardDuty usuário.

- Para API obter detalhes, consulte [DeleteFilter](#) na Referência de AWS CLI Comandos.

disable-organization-admin-account

O código de exemplo a seguir mostra como usar `disable-organization-admin-account`.

AWS CLI

Para remover uma conta como administrador delegado GuardDuty da sua organização

Este exemplo mostra como remover uma conta como administrador delegado do. GuardDuty

```
aws guardduty disable-organization-admin-account \  
  --admin-account-id 111122223333
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas com AWS organizações](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [DisableOrganizationAdminAccount](#) na Referência de AWS CLI Comandos.

disassociate-from-master-account

O código de exemplo a seguir mostra como usar `disassociate-from-master-account`.

AWS CLI

Para se desassociar da sua conta principal atual na região atual

O `disassociate-from-master-account` exemplo a seguir dissocia sua conta da conta GuardDuty principal atual na região atual. AWS

```
aws guardduty disassociate-from-master-account \  
  --detector-id d4b040365221be2b54a6264dcexample
```

Este comando não produz saída.

Para obter mais informações, consulte [Entendendo a relação entre contas GuardDuty principais e de membros](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [DisassociateFromMasterAccount](#) na Referência de AWS CLI Comandos.

get-detector

O código de exemplo a seguir mostra como usar `get-detector`.

AWS CLI

Para recuperar detalhes de um detector específico

O `get-detector` exemplo a seguir exibe os detalhes das configurações do detector especificado.

```
aws guardduty get-detector \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Conceitos e terminologia](#) no Guia do GuardDuty usuário.

- Para API obter detalhes, consulte [GetDetector](#) na Referência de AWS CLI Comandos.

get-findings

O código de exemplo a seguir mostra como usar `get-findings`.

AWS CLI

Exemplo 1: Para recuperar os detalhes de uma descoberta específica

O `get-findings` exemplo a seguir recupera os detalhes completos JSON da descoberta especificada.

```
aws guardduty get-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-id 1ab92989eaf0e742df4a014d5example
```

Saída:

```
{  
  "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"
}
]
}

```

Para obter mais informações, consulte [as descobertas](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [GetFindings](#) na Referência de AWS CLI Comandos.

get-ip-set

O código de exemplo a seguir mostra como usar `get-ip-set`.

AWS CLI

Para listar, obtenha detalhes sobre um conjunto de IP confiável especificado.

O `get-ip-set` exemplo a seguir mostra o status e os detalhes do conjunto de IP confiável especificado.

```
aws guardduty get-ip-set \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --ip-set-id d4b94fc952d6912b8f3060768example
```

Saída:

```
{  
  "Status": "ACTIVE",  
  "Location": "s3://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/  
customlist.csv",  
  "Tags": {},  
  "Format": "TXT",  
  "Name": "test-ip-set"  
}
```

Para obter mais informações, consulte [Trabalhando com listas de IPs confiáveis e listas de ameaças](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [GetIpSet](#) na Referência de AWS CLI Comandos.

get-master-account

O código de exemplo a seguir mostra como usar `get-master-account`.

AWS CLI

Para recuperar detalhes sobre sua conta principal na região atual

O `get-master-account` exemplo a seguir exibe o status e os detalhes da conta principal associada ao seu detector na região atual.

```
aws guardduty get-master-account \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Saída:

```
{  
  "Master": {  
    "InvitationId": "04b94d9704854a73f94e061e8example",
```

```
"InvitedAt": "2020-06-09T22:23:04.970Z",
"RelationshipStatus": "Enabled",
"AccountId": "123456789111"
}
}
```

Para obter mais informações, consulte [Entendendo a relação entre contas GuardDuty principais e de membros](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [GetMasterAccount](#) na Referência de AWS CLI Comandos.

list-detectors

O código de exemplo a seguir mostra como usar `list-detectors`.

AWS CLI

Para listar os detectores disponíveis na região atual

O `list-detectors` exemplo a seguir lista os detectores disponíveis na sua AWS região atual.

```
aws guardduty list-detectors
```

Saída:

```
{
  "DetectorIds": [
    "12abc34d567e8fa901bc2d34eexample"
  ]
}
```

Para obter mais informações, consulte [Conceitos e terminologia](#) no Guia do GuardDuty usuário.

- Para API obter detalhes, consulte [ListDetectors](#) na Referência de AWS CLI Comandos.

list-findings

O código de exemplo a seguir mostra como usar `list-findings`.

AWS CLI

Exemplo 1: Para listar todas as descobertas da região atual

O `list-findings` exemplo a seguir exibe uma lista de todas as `findingIds` da região atual, classificadas por severidade, da mais alta para a menor.

```
aws guardduty list-findings \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --sort-criteria '{"AttributeName": "severity", "OrderBy": "DESC"}'
```

Saída:

```
{
  "FindingIds": [
    "04b8ab50fd29c64fc771b232dexample",
    "5ab8ab50fd21373735c826d3aexample",
    "90b93de7aba69107f05bbe60bexample",
    ...
  ]
}
```

Para obter mais informações, consulte [as descobertas](#) no Guia GuardDuty do usuário.

Exemplo 2: Para listar descobertas da região atual que correspondam a um critério de descoberta específico

O `list-findings` exemplo a seguir exibe uma lista de todas as `findingIds` que correspondem a um tipo de descoberta especificado.

```
aws guardduty list-findings \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --finding-criteria '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/SSHBruteForce"]}}}'
```

Saída:

```
{
  "FindingIds": [
    "90b93de7aba69107f05bbe60bexample",
    "6eb9430d7023d30774d6f05e3example",
    "2eb91a2d060ac9a21963a5848example",
    "44b8ab50fd2b0039a9e48f570example",
    "9eb8ab4cd2b7e5b66ba4f5e96example",
    "e0b8ab3a38e9b0312cc390ceeexample"
  ]
}
```



```
}
```

Para obter mais informações, consulte [as descobertas](#) no Guia GuardDuty do usuário.

Exemplo 3: Para listar descobertas da região atual que correspondam a um conjunto específico de critérios de busca definidos em um JSON arquivo

O `list-findings` exemplo a seguir exibe uma lista de todos os `findingIds` que não estão arquivados e envolve o IAM usuário chamado “testuser”, conforme especificado em um JSON arquivo.

```
aws guardduty list-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-criteria file://myfile.json
```

Conteúdo de `myfile.json`:

```
{  
  "Criterion": {  
    "resource.accessKeyDetails.userName": {  
      "Eq": [  
        "testuser"  
      ]  
    },  
    "service.archived": {  
      "Eq": [  
        "false"  
      ]  
    }  
  }  
}
```

Saída:

```
{  
  "FindingIds": [  
    "1ab92989eaf0e742df4a014d5example"  
  ]  
}
```

Para obter mais informações, consulte [as descobertas](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [ListFindings](#) na Referência de AWS CLI Comandos.

list-invitations

O código de exemplo a seguir mostra como usar `list-invitations`.

AWS CLI

Para listar detalhes sobre seus convites para se tornar uma conta membro na região atual

O `list-invitations` exemplo a seguir lista detalhes e status de seus convites para se tornar uma conta GuardDuty membro na região atual.

```
aws guardduty list-invitations
```

Saída:

```
{
  "Invitations": [
    {
      "InvitationId": "d6b94fb03a66ff665f7db8764example",
      "InvitedAt": "2020-06-10T17:56:38.221Z",
      "RelationshipStatus": "Invited",
      "AccountId": "123456789111"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de GuardDuty contas por convite](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [ListInvitations](#) na Referência de AWS CLI Comandos.

list-ip-sets

O código de exemplo a seguir mostra como usar `list-ip-sets`.

AWS CLI

Para listar conjuntos de IP confiáveis na região atual

O `list-ip-sets` exemplo a seguir lista os conjuntos de IP confiáveis na sua AWS região atual.

```
aws guardduty list-ip-sets \
```

```
--detector-id 12abc34d567e8fa901bc2d34eexample
```

Saída:

```
{
  "IpSetIds": [
    "d4b94fc952d6912b8f3060768example"
  ]
}
```

Para obter mais informações, consulte [Trabalhando com listas de IPs confiáveis e listas de ameaças](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [ListIpSets](#) na Referência de AWS CLI Comandos.

list-members

O código de exemplo a seguir mostra como usar `list-members`.

AWS CLI

Para listar todos os membros na região atual

O `list-members` exemplo a seguir lista todas as contas dos membros e seus detalhes para a região atual.

```
aws guardduty list-members \
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [Entendendo a relação entre contas GuardDuty principais e de membros](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [ListMembers](#) na Referência de AWS CLI Comandos.

update-ip-set

O código de exemplo a seguir mostra como usar `update-ip-set`.

AWS CLI

Para atualizar um conjunto de IP confiável

O `update-ip-set` exemplo a seguir mostra como atualizar os detalhes de um conjunto de IP confiável.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com listas de IPs confiáveis e listas de ameaças](#) no Guia GuardDuty do usuário.

- Para API obter detalhes, consulte [UpdateIpSet](#) na Referência de AWS CLI Comandos.

AWS Health exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Health.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-affected-entities

O código de exemplo a seguir mostra como usar `describe-affected-entities`.

AWS CLI

Para listar as entidades afetadas por um evento AWS Health especificado

O `describe-affected-entities` exemplo a seguir lista as entidades afetadas pelo evento AWS Health especificado. Esse evento é uma notificação de cobrança para a AWS conta.

```
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
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Registro de eventos](#) no AWS Health User Guide.

- Para API obter detalhes, consulte [DescribeAffectedEntities](#) na Referência de AWS CLI Comandos.

describe-event-details

O código de exemplo a seguir mostra como usar `describe-event-details`.

AWS CLI

Para listar informações sobre um evento AWS de Saúde

O `describe-event-details` exemplo a seguir lista informações sobre o evento AWS Health especificado.

```
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
```

Saída:

```
{
  "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": []
}
```

Para obter mais informações, consulte o [painel Detalhes do evento](#) no AWS Health User Guide.

- Para API obter detalhes, consulte [DescribeEventDetails](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar describe-events.

AWS CLI

Exemplo 1: Para listar eventos de AWS Health

O describe-events exemplo a seguir lista eventos recentes AWS de Health.

```
aws health describe-events \
  --region us-east-1
```

Saída:

```
{
  "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_EXAMPLE1111",
      "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"
    }
  ]
}

```

Para obter mais informações, consulte [Introdução ao AWS Personal Health Dashboard](#) no AWS Health User Guide.

Exemplo 2: Para listar eventos de AWS Health por serviço e código de status do evento

O `describe-events` exemplo a seguir lista eventos de AWS saúde para o Amazon Elastic Compute Cloud (AmazonEC2) em que o status do evento está fechado.

```

aws health describe-events \
  --filter "services=EC2,eventStatusCodes=closed"

```

Saída:

```
{
  "events": [
    {
      "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_VKTXI_EXAMPLE111",
      "service": "EC2",
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
      "eventTypeCategory": "issue",
      "region": "us-east-1",
      "startTime": 1587462325.096,
      "endTime": 1587464204.774,
      "lastUpdatedTime": 1587464204.865,
      "statusCode": "closed",
      "eventScopeCode": "PUBLIC"
    },
    {
      "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
      "service": "EC2",
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
      "eventTypeCategory": "issue",
      "region": "us-east-1",
      "startTime": 1586473044.284,
      "endTime": 1586479706.091,
      "lastUpdatedTime": 1586479706.153,
      "statusCode": "closed",
      "eventScopeCode": "PUBLIC"
    },
    {
      "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
      "service": "EC2",
      "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
      "eventTypeCategory": "issue",
      "region": "ap-southeast-2",
      "startTime": 1583881487.483,
      "endTime": 1583885056.785,
      "lastUpdatedTime": 1583885057.052,
      "statusCode": "closed",
      "eventScopeCode": "PUBLIC"
    }
  ]
}
```

Para obter mais informações, consulte [Introdução ao AWS Personal Health Dashboard](#) no AWS Health User Guide.

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

HealthImaging exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with HealthImaging.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

copy-image-set

O código de exemplo a seguir mostra como usar `copy-image-set`.

AWS CLI

Exemplo 1: para copiar um conjunto de imagens sem um destino.

O exemplo de código `copy-image-set` a seguir faz uma cópia duplicada de uma imagem sem um destino.

```
aws medical-imaging copy-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

Saída:

```
{
  "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"
}
```

Exemplo 2: para copiar um conjunto de imagens com um destino.

O exemplo de código `copy-image-set` a seguir faz uma cópia duplicada de um conjunto de imagens com um destino.

```
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"} }'
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Copiar um conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [CopyImageSet](#) na Referência de AWS CLI Comandos.

create-datastore

O código de exemplo a seguir mostra como usar create-datastore.

AWS CLI

Para criar um armazenamento de dados

O exemplo de código create-datastore a seguir cria um armazenamento de dados com o nome de my-datastore.

```
aws medical-imaging create-datastore \
  --datastore-name "my-datastore"
```

Saída:

```
{
  "datastoreId": "12345678901234567890123456789012",
  "datastoreStatus": "CREATING"
}
```

Para obter mais informações, consulte [Criação de um armazenamento de dados](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [CreateDatastore](#) na Referência de AWS CLI Comandos.

delete-datastore

O código de exemplo a seguir mostra como usar delete-datastore.

AWS CLI

Para excluir um armazenamento de dados

O exemplo de código delete-datastore a seguir exclui um armazenamento de dados.

```
aws medical-imaging delete-datastore \  
  --datastore-id "12345678901234567890123456789012"
```

Saída:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "datastoreStatus": "DELETING"  
}
```

Para obter mais informações, consulte [Excluindo um armazenamento de dados](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [DeleteDatastore](#) na Referência de AWS CLI Comandos.

delete-image-set

O código de exemplo a seguir mostra como usar delete-image-set.

AWS CLI

Para excluir um conjunto de imagens

O exemplo de código delete-image-set a seguir exclui um conjunto de imagens.

```
aws medical-imaging delete-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Saída:

```
{
```

```
"imageSetWorkflowStatus": "DELETING",
"imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
"imageSetState": "LOCKED",
"datastoreId": "12345678901234567890123456789012"
}
```

Para obter mais informações, consulte [Excluir um conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [DeletImageSet](#) na Referência de AWS CLI Comandos.

get-datastore

O código de exemplo a seguir mostra como usar `get-datastore`.

AWS CLI

Para obter as propriedades de um armazenamento de dados

O exemplo de código `get-datastore` a seguir obtém as propriedades de um armazenamento de dados.

```
aws medical-imaging get-datastore \
  --datastore-id 12345678901234567890123456789012
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Obter propriedades do armazenamento de dados](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [GetDatastore](#) na Referência de AWS CLI Comandos.

get-dicom-import-job

O código de exemplo a seguir mostra como usar `get-dicom-import-job`.

AWS CLI

Para obter as propriedades de um trabalho de importação dicom

O exemplo de código `get-dicom-import-job` a seguir obtém as propriedades de um trabalho de importação dicom.

```
aws medical-imaging get-dicom-import-job \  
  --datastore-id "12345678901234567890123456789012" \  
  --job-id "09876543210987654321098765432109"
```

Saída:

```
{  
  "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/"  
  }  
}
```

Para obter mais informações, consulte [Obter propriedades do trabalho de importação](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [GetDICOMImport Job](#) in AWS CLI Command Reference.

get-image-frame

O código de exemplo a seguir mostra como usar `get-image-frame`.

AWS CLI

Para obter dados de pixels do conjunto de imagens

O exemplo de código `get-image-frame` a seguir obtém um quadro de imagem.

```
aws medical-imaging get-image-frame \  
  --datastore-id "12345678901234567890123456789012" \  
  --image-set-id "98765412345612345678907890789012" \  
  --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \  
  imageframe.jph
```

Observação: esse exemplo de código não inclui a saída porque a `GetImageFrame` ação retorna um fluxo de dados de pixels para o arquivo `imageframe.jph`. Para obter informações sobre decodificação e visualização de quadros de imagem, consulte bibliotecas de HTJ2K decodificação.

Para obter mais informações, consulte [Obter dados de pixels do conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [GetImageFrame](#) na Referência de AWS CLI Comandos.

`get-image-set-metadata`

O código de exemplo a seguir mostra como usar `get-image-set-metadata`.

AWS CLI

Exemplo 1: para obter os metadados de um conjunto de imagens sem versão

O exemplo de código `get-image-set-metadata` a seguir obtém metadados para um conjunto de imagens sem especificar uma versão.

Observação: `outfile` é um parâmetro obrigatório

```
aws medical-imaging get-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  studymetadata.json.gz
```

Os metadados retornados são compactados com o gzip e armazenados no arquivo `studymetadata.json.gz`. Para visualizar o conteúdo do JSON objeto retornado, você deve primeiro descompactá-lo.

Saída:

```
{
  "contentType": "application/json",
  "contentEncoding": "gzip"
}
```

Exemplo 2: para obter os metadados de um conjunto de imagens com versão

O exemplo de código `get-image-set-metadata` a seguir obtém metadados para um conjunto de imagens com uma versão especificada.

Observação: `outfile` é um parâmetro obrigatório

```
aws medical-imaging get-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --version-id 1 \
  studymetadata.json.gz
```

Os metadados retornados são compactados com o gzip e armazenados no arquivo `studymetadata.json.gz`. Para visualizar o conteúdo do JSON objeto retornado, você deve primeiro descompactá-lo.

Saída:

```
{
  "contentType": "application/json",
  "contentEncoding": "gzip"
}
```

Para obter mais informações, consulte [Obter metadados do conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [GetImageSetMetadata](#) na Referência de AWS CLI Comandos.

get-image-set

O código de exemplo a seguir mostra como usar `get-image-set`.

AWS CLI

Para obter as propriedades do conjunto de imagens

O exemplo de código `get-image-set` a seguir obtém as propriedades de um conjunto de imagens.

```
aws medical-imaging get-image-set \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id 18f88ac7870584f58d56256646b4d92b \
  --version-id 1
```

Saída:

```
{
  "versionId": "1",
  "imageSetWorkflowStatus": "COPIED",
  "updatedAt": 1680027253.471,
  "imageSetId": "18f88ac7870584f58d56256646b4d92b",
  "imageSetState": "ACTIVE",
  "createdAt": 1679592510.753,
  "datastoreId": "12345678901234567890123456789012"
}
```

Para obter mais informações, consulte [Como obter propriedades do conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [GetImageSet](#) na Referência de AWS CLI Comandos.

list-datastores

O código de exemplo a seguir mostra como usar `list-datastores`.

AWS CLI

Para listar armazenamentos de dados

O exemplo de código `list-datastores` a seguir lista os armazenamentos de dados disponíveis.

```
aws medical-imaging list-datastores
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Listar armazenamentos de dados](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [ListDatastores](#) na Referência de AWS CLI Comandos.

list-dicom-import-jobs

O código de exemplo a seguir mostra como usar `list-dicom-import-jobs`.

AWS CLI

Para listar trabalhos de importação dicom

O exemplo de código `list-dicom-import-jobs` a seguir lista os trabalhos de importação dicom.

```
aws medical-imaging list-dicom-import-jobs \
  --datastore-id "12345678901234567890123456789012"
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Listar trabalhos de importação](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [ListDICOMImport Jobs](#) in AWS CLI Command Reference.

list-image-set-versions

O código de exemplo a seguir mostra como usar `list-image-set-versions`.

AWS CLI

Para listar as versões de um conjunto de imagens

O exemplo de código `list-image-set-versions` a seguir lista o histórico de versões de um conjunto de imagens.

```

aws medical-imaging list-image-set-versions \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e

```

Saída:

```

{
  "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
  }
]
}

```

Para obter mais informações, consulte [Listar as versões do conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [ListImageSetVersions](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Exemplo 1: para listar as tags de recurso de um armazenamento de dados

O exemplo de código `list-tags-for-resource` a seguir lista as tags de um armazenamento de dados.

```
aws medical-imaging list-tags-for-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012"
```

Saída:

```
{  
  "tags":{  
    "Deployment":"Development"  
  }  
}
```

Exemplo 2: para listar tags de recurso de um conjunto de imagens

O exemplo de código `list-tags-for-resource` a seguir lista as tags de um conjunto de imagens.

```
aws medical-imaging list-tags-for-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b"
```

Saída:

```
{  
  "tags":{  
    "Deployment":"Development"  
  }  
}
```

Para obter mais informações, consulte Como [marcar recursos AWS HealthImaging](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

search-image-sets

O código de exemplo a seguir mostra como usar `search-image-sets`.

AWS CLI

Exemplo 1: Para pesquisar conjuntos de imagens com um EQUAL operador

O exemplo de `search-image-sets` código a seguir usa o EQUAL operador para pesquisar conjuntos de imagens com base em um valor específico.

```
aws medical-imaging search-image-sets \  
  --datastore-id 12345678901234567890123456789012 \  
  --search-criteria file://search-criteria.json
```

Conteúdo de `search-criteria.json`

```
{  
  "filters": [{  
    "values": [{"DICOMPatientId" : "SUBJECT08701"}],  
    "operator": "EQUAL"  
  }]  
}
```

Saída:

```
{  
  "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"  
  }]  
}
```

```
}
```

Exemplo 2: Para pesquisar conjuntos de imagens com um BETWEEN operador usando DICOMStudyDate e DICOMStudyTime

O exemplo de `search-image-sets` código a seguir pesquisa conjuntos de imagens com DICOM estudos gerados entre 1º de janeiro de 1990 (00h) e 1º de janeiro de 2023 (00h).

Nota: `DICOMStudyTime` é opcional. Se não estiver presente, 00h (início do dia) é o valor da hora para as datas fornecidas para filtragem.

```
aws medical-imaging search-image-sets \  
  --datastore-id 12345678901234567890123456789012 \  
  --search-criteria file://search-criteria.json
```

Conteúdo de `search-criteria.json`

```
{  
  "filters": [{  
    "values": [{  
      "DICOMStudyDateAndTime": {  
        "DICOMStudyDate": "19900101",  
        "DICOMStudyTime": "000000"  
      }  
    },  
    {  
      "DICOMStudyDateAndTime": {  
        "DICOMStudyDate": "20230101",  
        "DICOMStudyTime": "000000"  
      }  
    }  
  ]},  
  "operator": "BETWEEN"  
}]  
}
```

Saída:

```
{  
  "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"
  ]
}

```

Exemplo 3: Para pesquisar conjuntos de imagens com um BETWEEN operador usando createdAt (os estudos de tempo foram persistidos anteriormente)

O exemplo de search-image-sets código a seguir pesquisa conjuntos de imagens com DICOM estudos persistentes HealthImaging entre os intervalos de tempo no fuso UTC horário.

Nota: Forneça um exemplo createdAt de formato ("1985-04-12T 23:20:50.52 Z").

```

aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json

```

Conteúdo de search-criteria.json

```

{
  "filters": [{
    "values": [{
      "createdAt": "1985-04-12T23:20:50.52Z"
    },
    {
      "createdAt": "2022-04-12T23:20:50.52Z"
    }
  ]],
  "operator": "BETWEEN"
}]

```

```
}

```

Saída:

```
{
  "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"
  }]
}
```

Exemplo 4: Para pesquisar conjuntos de imagens com um EQUAL operador ligado DICOMSeriesInstanceUID e BETWEEN ligado updatedAt e classificar a resposta em ASC ordem no updatedAt campo

O exemplo de `search-image-sets` código a seguir pesquisa conjuntos de imagens com um EQUAL operador ligado DICOMSeriesInstanceUID e BETWEEN ligado updatedAt e classifica a resposta em ASC ordem no updatedAt campo.

Nota: Forneça um exemplo updatedAt de formato (“1985-04-12T 23:20:50.52 Z”).

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json
```

Conteúdo de `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"
  }
}
```

Saída:

```
{
  "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"
  }]
}
```

```
}
```

Para obter mais informações, consulte [Pesquisar conjuntos de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [SearchImageSets](#) na Referência de AWS CLI Comandos.

start-dicom-import-job

O código de exemplo a seguir mostra como usar `start-dicom-import-job`.

AWS CLI

Para iniciar um trabalho de importação dicom

O exemplo de código `start-dicom-import-job` a seguir inicia um trabalho de importação 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"
```

Saída:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "jobId": "09876543210987654321098765432109",  
  "jobStatus": "SUBMITTED",  
  "submittedAt": "2022-08-12T11:28:11.152000+00:00"  
}
```

Para obter mais informações, consulte [Iniciando um trabalho de importação](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [StartDICOMImport Job](#) in AWS CLI Command Reference.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Exemplo 1: para marcar um armazenamento de dados

Os exemplos de código `tag-resource` a seguir marcam um armazenamento de dados.

```
aws medical-imaging tag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tags '{"Deployment":"Development"}'
```

Este comando não produz saída.

Exemplo 2: para marcar um conjunto de imagens

Os exemplos de código `tag-resource` a seguir marcam um conjunto de imagens.

```
aws medical-imaging tag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tags '{"Deployment":"Development"}'
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar recursos AWS HealthImaging](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Exemplo 1: para desmarcar um armazenamento de dados

O exemplo de código `untag-resource` a seguir desmarca um armazenamento de dados.

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tags '{"Deployment":"Development"}'
```

```
--tag-keys '["Deployment"]'
```

Este comando não produz saída.

Exemplo 2: para desmarcar um conjunto de imagens

O exemplo de código `untag-resource` a seguir desmarca um conjunto de imagens.

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tag-keys '["Deployment"]'
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar recursos AWS HealthImaging](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-image-set-metadata

O código de exemplo a seguir mostra como usar `update-image-set-metadata`.

AWS CLI

Para inserir ou atualizar um atributo nos metadados do conjunto de imagens

O exemplo de `update-image-set-metadata` código a seguir insere ou atualiza um atributo nos metadados do conjunto de imagens.

```
aws medical-imaging update-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92bd0d8838c72a3f25d00d13616f87e \  
  --latest-version-id 1 \  
  --update-image-set-metadata-updates file://metadata-updates.json
```

Conteúdo de `metadata-updates.json`

```
{
```



```

    "DICOMUpdates": {
      "updateableAttributes":
"eyJTY2h1bWFWZXJzaW9uIjoxLjEsIlBhdGllbnQiOnsiRElDT00iOnsiUGF0aWVudE5hbWUiOiJNWF5NWCJ9fX0="
    }
  }
}

```

Nota: updateableAttributes é uma string codificada JSON em Base64. Aqui está a string não codificada. JSON

```
{ "SchemaVersion": "1.1", "Paciente": { "DICOM": { "PatientName": "MX^MX" } } }
```

Saída:

```

{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}

```

Para remover um atributo dos metadados do conjunto de imagens

O exemplo de update-image-set-metadata código a seguir remove um atributo dos metadados do conjunto de imagens.

```

aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --latest-version-id 1 \
  --update-image-set-metadata-updates file://metadata-updates.json

```

Conteúdo de metadata-updates.json

```

{
  "DICOMUpdates": {
    "removableAttributes":
"e1NjaGVtYVZlcnNpb246MS4xLFN0dWR5OntESUNPTTp7U3R1ZH1EZnJcm1wdG1vbjpdSEVTVH19fQo="
  }
}

```

```
}

```

Nota: `removableAttributes` é uma string codificada JSON em Base64. Aqui está a string não codificada. JSON A chave e o valor devem corresponder ao atributo a ser removido.

```
{ "SchemaVersion": "1.1", "Estudo": { "DICOM": { "StudyDescription": "CHEST" } } }
```

Saída:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}
```

Para remover uma instância dos metadados do conjunto de imagens

O exemplo de `update-image-set-metadata` código a seguir remove uma instância dos metadados do conjunto de imagens.

```
aws medical-imaging update-image-set-metadata \
  --datastore-id 12345678901234567890123456789012 \
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
  --latest-version-id 1 \
  --update-image-set-metadata-updates file://metadata-updates.json
```

Conteúdo de `metadata-updates.json`

```
{
  "DICOMUpdates": {
    "removableAttributes":
      "eezEuMS4xLjEuMS4xLjEyMzQ1LjEyMzQ1Njc4OTAxMi4xMjMuMTIzNDU2Nzg5MDEyMzQuMTp7SW5zdGFuY2VzOnsxL"
  }
}
```

Nota: `removableAttributes` é uma string codificada JSON em Base64. Aqui está a string não codificada. JSON

```
{"1.1.1.1.1.12345.123456789012.123.12345678901234.1": {"Instâncias":  
{"1.1.1.1.1.1.1.1.12345.123456789012.123.12345678901234.1": {}}}}
```

Saída:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",  
  "updatedAt": 1680042257.908,  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "createdAt": 1680027126.436,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

Para obter mais informações, consulte [Atualização dos metadados do conjunto de imagens](#) no Guia do AWS HealthImaging desenvolvedor.

- Para API obter detalhes, consulte [UpdateImageSetMetadata](#) na Referência de AWS CLI Comandos.

HealthLake exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with HealthLake.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-fhir-datastore

O código de exemplo a seguir mostra como usar `create-fhir-datastore`.

AWS CLI

Para criar um armazenamento FHIR de dados.

O `create-fhir-datastore` exemplo a seguir demonstra como criar um novo armazenamento de dados na Amazon HealthLake.

```
aws healthlake create-fhir-datastore \  
  --region us-east-1 \  
  --datastore-type-version R4 \  
  --datastore-version R4 \  
  --datastore-name "FhirTestDatastore"
```

Saída:

```
{  
  "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)"  
}
```

Para obter mais informações, consulte [Criação e monitoramento de um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [CreateFhirDatastore](#) na Referência de AWS CLI Comandos.

delete-fhir-datastore

O código de exemplo a seguir mostra como usar `delete-fhir-datastore`.

AWS CLI

Para excluir um armazenamento FHIR de dados

O `delete-fhir-datastore` exemplo a seguir demonstra como excluir um armazenamento de dados e todo o seu conteúdo na Amazon HealthLake.

```
aws healthlake delete-fhir-datastore \  
  --datastore-id (Datastore ID)
```

```
--datastore-id (Data Store ID) \  
--region us-east-1
```

Saída:

```
{  
  "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)"  
}
```

Para obter mais informações, consulte Criação e monitoramento de um armazenamento de FHIR dados < <https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-healthlake.html> > no Amazon Developer Guide. FHIR HealthLake

- Para API obter detalhes, consulte [DeleteFhirDatastore](#) na Referência de AWS CLI Comandos.

describe-fhir-datastore

O código de exemplo a seguir mostra como usar `describe-fhir-datastore`.

AWS CLI

Para descrever um armazenamento FHIR de dados

O `describe-fhir-datastore` exemplo a seguir demonstra como encontrar as propriedades de um armazenamento de dados na Amazon HealthLake.

```
aws healthlake describe-fhir-datastore \  
--datastore-id "1f2f459836ac6c513ce899f9e4f66a59" \  
--region us-east-1
```

Saída:

```
{  
  "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)"
  }
}

```

Para obter mais informações, consulte [Criação e monitoramento de um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [DescribeFhirDatastore](#) na Referência de AWS CLI Comandos.

describe-fhir-export-job

O código de exemplo a seguir mostra como usar `describe-fhir-export-job`.

AWS CLI

Para descrever um trabalho FHIR de exportação

O `describe-fhir-export-job` exemplo a seguir mostra como encontrar as propriedades de um trabalho de FHIR exportação na Amazon HealthLake.

```

aws healthlake describe-fhir-export-job \
  --datastore-id (Datastore ID) \
  --job-id 9b9a51943afaedd0a8c0c26c49135a31

```

Saída:

```

{
  "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)"
  }
}

```

Para obter mais informações, consulte [Exportação de arquivos de um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [DescribeFhirExportJob](#) na Referência de AWS CLI Comandos.

describe-fhir-import-job

O código de exemplo a seguir mostra como usar `describe-fhir-import-job`.

AWS CLI

Para descrever um trabalho FHIR de importação

O `describe-fhir-import-job` exemplo a seguir mostra como conhecer as propriedades de um trabalho de FHIR importação usando a Amazon HealthLake.

```

aws healthlake describe-fhir-import-job \
  --datastore-id (Datastore ID) \
  --job-id c145fbb27b192af392f8ce6e7838e34f \
  --region us-east-1

```

Saída:

```

{
  "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)"
  }
}

```

Para obter mais informações, consulte [Importação de arquivos para um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [DescribeFhirImportJob](#) na Referência de AWS CLI Comandos.

list-fhir-datastores

O código de exemplo a seguir mostra como usar `list-fhir-datastores`.

AWS CLI

Para listar armazenamentos FHIR de dados

O `list-fhir-datastores` exemplo a seguir mostra como usar o comando e como os usuários podem filtrar os resultados com base no status do Data Store na Amazon HealthLake.

```

aws healthlake list-fhir-datastores \
  --region us-east-1 \
  --filter DatastoreId=ACTIVE

```

Saída:

```

{
  "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>"
  }
]
}

```

Para obter mais informações, consulte [Criação e monitoramento de um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [ListFhirDatastores](#) na Referência de AWS CLI Comandos.

list-fhir-export-jobs

O código de exemplo a seguir mostra como usar `list-fhir-export-jobs`.

AWS CLI

Para listar todos os trabalhos FHIR de exportação

O `list-fhir-export-jobs` exemplo a seguir mostra como usar o comando para visualizar uma lista de trabalhos de exportação associados a uma conta.

```

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)

```

Saída:

```
{
  "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
```

Para obter mais informações, consulte [Exportação de arquivos de um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [ListFhirExportJobs](#) na Referência de AWS CLI Comandos.

list-fhir-import-jobs

O código de exemplo a seguir mostra como usar `list-fhir-import-jobs`.

AWS CLI

Para listar todos os trabalhos FHIR de importação

O `list-fhir-import-jobs` exemplo a seguir mostra como usar o comando para visualizar uma lista de todos os trabalhos de importação associados a uma conta.

```
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)
```

Saída:

```
{  
  "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
```

Para obter mais informações, consulte [Importação de arquivos para o FHIR Data Store](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [ListFhirImportJobs](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um armazenamento de dados

O `list-tags-for-resource` exemplo a seguir lista as tags associadas ao armazenamento de dados especificado. :

```
aws healthlake list-tags-for-resource \  

```

```
--resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/  
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \  
--region us-east-1
```

Saída:

```
{  
  "tags": {  
    "key": "value",  
    "key1": "value1"  
  }  
}
```

Para obter mais informações, consulte [Marcar recursos na Amazon HealthLake no Amazon HealthLake Developer Guide](#).

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

start-fhir-export-job

O código de exemplo a seguir mostra como usar `start-fhir-export-job`.

AWS CLI

Para iniciar um trabalho FHIR de exportação

O `start-fhir-export-job` exemplo a seguir mostra como iniciar um trabalho de FHIR exportação usando a 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)
```

Saída:

```
{  
  "DatastoreId": "(Datastore ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "9b9a51943afaedd0a8c0c26c49135a31"
```

```
}
```

Para obter mais informações, consulte [Exportação de arquivos de um armazenamento de FHIR dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [StartFhirExportJob](#) na Referência de AWS CLI Comandos.

start-fhir-import-job

O código de exemplo a seguir mostra como usar `start-fhir-import-job`.

AWS CLI

Para iniciar um trabalho FHIR de importação

O `start-fhir-import-job` exemplo a seguir mostra como iniciar um trabalho de FHIR importação usando a 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
```

Saída:

```
{
  "DatastoreId": "(Datastore ID)",
  "JobStatus": "SUBMITTED",
  "JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

Para obter mais informações, consulte [Importação de arquivos para um armazenamento de FHIR dados](#) em <https://docs.aws.amazon.com/healthlake/latest/devguide/import-datastore.html> no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [StartFhirImportJob](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag ao Data Store

O `tag-resource` exemplo a seguir mostra como adicionar uma tag a um armazenamento de dados.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte “Adicionar uma tag a um armazenamento de dados < <https://docs.aws.amazon.com/healthlake/latest/devguide/add-a-tag.html>>” no Amazon Developer Guide. HealthLake .

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um armazenamento de dados.

O `untag-resource` exemplo a seguir mostra como remover tags de um armazenamento de dados.

```
aws healthlake untag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/fhir/  
b91723d65c6fdeb1d26543a49d2ed1fa" \  
  --tag-keys ["key1"] \  
  --region us-east-1
```

Este comando não produz saída.

Para obter mais informações, consulte [Remoção de tags de um armazenamento de dados](#) no Amazon HealthLake Developer Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

HealthOmics exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with HealthOmics.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

abort-multipart-read-set-upload

O código de exemplo a seguir mostra como usar `abort-multipart-read-set-upload`.

AWS CLI

Para interromper o upload de um conjunto de leitura em várias partes

O `abort-multipart-read-set-upload` exemplo a seguir interrompe o upload de um conjunto de leitura de várias partes em seu armazenamento de HealthOmics sequências.

```
aws omics abort-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455
```

Este comando não produz saída.

Para obter mais informações, consulte [Carregamento direto para um armazenamento de sequências](#) no Guia AWS HealthOmics do usuário.

- Para API obter detalhes, consulte [AbortMultipartReadSetUpload](#) na Referência de AWS CLI Comandos.

accept-share

O código de exemplo a seguir mostra como usar `accept-share`.

AWS CLI

Para aceitar um compartilhamento dos dados do Analytics, armazene

O `accept-share` exemplo a seguir aceita um compartilhamento dos dados do HealthOmics Analytics Store.

```
aws omics accept-share \  
  ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Saída:

```
{  
  "status": "ACTIVATING"  
}
```

Para obter mais informações, consulte [Compartilhamento entre contas](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [AcceptShare](#) na Referência de AWS CLI Comandos.

batch-delete-read-set

O código de exemplo a seguir mostra como usar `batch-delete-read-set`.

AWS CLI

Para excluir vários conjuntos de leitura

O `batch-delete-read-set` exemplo a seguir exclui dois conjuntos de leitura.

```
aws omics batch-delete-read-set \  
  --sequence-store-id 1234567890 \  
  --read-set-id 1234567890
```



```
--ids 1234567890 0123456789
```

Se houver um erro ao excluir qualquer um dos conjuntos de leitura especificados, o serviço retornará uma lista de erros.

```
{
  "errors": [
    {
      "code": "",
      "id": "0123456789",
      "message": "The specified readset does not exist."
    }
  ]
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [BatchDeleteReadSet](#) na Referência de AWS CLI Comandos.

cancel-annotation-import-job

O código de exemplo a seguir mostra como usar `cancel-annotation-import-job`.

AWS CLI

Para cancelar um trabalho de importação de anotações

O `cancel-annotation-import-job` exemplo a seguir cancela um trabalho de importação de anotações com ID. `04f57618-xmpl-4fd0-9349-e5a85aefb997`

```
aws omics cancel-annotation-import-job \
  --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CancelAnnotationImportJob](#) na Referência de AWS CLI Comandos.

cancel-run

O código de exemplo a seguir mostra como usar `cancel-run`.

AWS CLI

Para cancelar uma corrida

O `cancel-run` exemplo a seguir cancela uma execução com ID1234567.

```
aws omics cancel-run \  
  --id 1234567
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CancelRun](#) na Referência de AWS CLI Comandos.

cancel-variant-import-job

O código de exemplo a seguir mostra como usar `cancel-variant-import-job`.

AWS CLI

Para cancelar um trabalho de importação de variantes

O `cancel-variant-import-job` exemplo a seguir cancela um trabalho de importação de variantes com ID69cb65d6-xmp1-4a4a-9025-4565794b684e.

```
aws omics cancel-variant-import-job \  
  --job-id 69cb65d6-xmp1-4a4a-9025-4565794b684e
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CancelVariantImportJob](#) na Referência de AWS CLI Comandos.

complete-multipart-read-set-upload

O código de exemplo a seguir mostra como usar `complete-multipart-read-set-upload`.

AWS CLI

Para concluir um upload de várias partes depois de fazer o upload de todos os componentes.

O `complete-multipart-read-set-upload` exemplo a seguir conclui um upload de várias partes em um armazenamento de sequências depois que todos os componentes tiverem sido carregados.

```
aws omics complete-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/  
Vobh5zBDb4W4=", "partNumber":1, "partSource":"SOURCE1"}]'
```

Saída:

```
{  
  "readSetId": "0000000001"  
  "readSetId": "0000000002"  
  "readSetId": "0000000003"  
}
```

Para obter mais informações, consulte [Carregamento direto para um armazenamento de sequências](#) no Guia AWS HealthOmics do usuário.

- Para API obter detalhes, consulte [CompleteMultipartReadSetUpload](#) na Referência de AWS CLI Comandos.

`create-annotation-store-version`

O código de exemplo a seguir mostra como usar `create-annotation-store-version`.

AWS CLI

Para criar uma nova versão de um repositório de anotações

O `create-annotation-store-version` exemplo a seguir cria uma nova versão de um repositório de anotações.

```
aws omics create-annotation-store-version \  
  --name my_annotation_store \  
  --version-name my_version
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Criação de novas versões de repositórios de anotações](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [CreateAnnotationStoreVersion](#) na Referência de AWS CLI Comandos.

create-annotation-store

O código de exemplo a seguir mostra como usar create-annotation-store.

AWS CLI

Exemplo 1: Para criar um repositório de VCF anotações

O create-annotation-store exemplo a seguir cria um repositório de anotações de VCF 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
```

Saída:

```
{
  "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"
}

```

Exemplo 2: Para criar um repositório de TSV anotações

O `create-annotation-store` exemplo a seguir cria um repositório de anotações de TSV 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 as opções de formato para anotações.

```

{
  "tsvStoreOptions": {
    "annotationType": "CHR_START_END_ZERO_BASE",
    "formatToHeader": {
      "CHR": "chromosome",
      "START": "start",
      "END": "end"
    },
    "schema": [
      {
        "chromosome": "STRING"
      },
      {
        "start": "LONG"
      },
      {
        "end": "LONG"
      },
      {
        "name": "STRING"
      }
    ]
  }
}

```

```
    ]  
  }  
}
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  }  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CreateAnnotationStore](#) na Referência de AWS CLI Comandos.

create-multipart-read-set-upload

O código de exemplo a seguir mostra como usar `create-multipart-read-set-upload`.

AWS CLI

Para iniciar um upload do conjunto de leitura em várias partes.

O `create-multipart-read-set-upload` exemplo a seguir inicia o upload de um conjunto de leitura em várias partes.

```
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"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Carregamento direto para um armazenamento de sequências](#) no Guia AWS HealthOmics do usuário.

- Para API obter detalhes, consulte [CreateMultipartReadSetUpload](#) na Referência de AWS CLI Comandos.

create-reference-store

O código de exemplo a seguir mostra como usar `create-reference-store`.

AWS CLI

Para criar uma loja de referência

O `create-reference-store` exemplo a seguir cria um repositório de referênciamy-ref-store.

```
aws omics create-reference-store \  
  --name my-ref-store
```

Saída:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",  
  "creationTime": "2022-11-22T22:13:25.947Z",  
  "id": "1234567890",  
  "name": "my-ref-store"  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CreateReferenceStore](#) na Referência de AWS CLI Comandos.

create-run-group

O código de exemplo a seguir mostra como usar `create-run-group`.

AWS CLI

Para criar um grupo de execução

O `create-run-group` exemplo a seguir cria um grupo de execução chamadocram-converter.

```
aws omics create-run-group \  
  --name cram-converter \  
  --
```



```
--max-cpus 20 \  
--max-duration 600
```

Saída:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",  
  "id": "1234567",  
  "tags": {}  
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CreateRunGroup](#) na Referência de AWS CLI Comandos.

create-sequence-store

O código de exemplo a seguir mostra como usar create-sequence-store.

AWS CLI

Para criar um armazenamento de sequências

O create-sequence-store exemplo a seguir cria um armazenamento de sequências.

```
aws omics create-sequence-store \  
--name my-seq-store
```

Saída:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",  
  "creationTime": "2022-11-23T01:24:33.629Z",  
  "id": "1234567890",  
  "name": "my-seq-store"  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CreateSequenceStore](#) na Referência de AWS CLI Comandos.

create-share

O código de exemplo a seguir mostra como usar create-share.

AWS CLI

Para criar um compartilhamento de uma loja de HealthOmics análises

O create-share exemplo a seguir mostra como criar um compartilhamento de uma loja de HealthOmics análises que pode ser aceito por um assinante fora da conta.

```
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"
```

Saída:

```
{  
  "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",  
  "name": "my_Share-123",  
  "status": "PENDING"  
}
```

Para obter mais informações, consulte [Compartilhamento entre contas](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [CreateShare](#) na Referência de AWS CLI Comandos.

create-variant-store

O código de exemplo a seguir mostra como usar create-variant-store.

AWS CLI

Para criar uma loja de variantes

O create-variant-store exemplo a seguir cria um armazenamento de variantes chamadomy_var_store.

```
aws omics create-variant-store \  

```

```
--name my_var_store \  
--reference referenceArn=arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CreateVariantStore](#) na Referência de AWS CLI Comandos.

create-workflow

O código de exemplo a seguir mostra como usar create-workflow.

AWS CLI

Para criar um fluxo de trabalho

O create-workflow exemplo a seguir cria um WDL fluxo de trabalho.

```
aws omics create-workflow \  
  --name cram-converter \  
  --engine WDL \  
  --definition-zip fileb://workflow-crambam.zip \  
  --parameter-template file://workflow-params.json
```

workflow-crambam.zip é um ZIP arquivo contendo uma definição de fluxo de trabalho.

workflow-params.json define parâmetros de tempo de execução para o fluxo de trabalho.

```
{
```

```
"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
}
}
```

Saída:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
  "id": "1234567",
  "status": "CREATING",
  "tags": {}
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [CreateWorkflow](#) na Referência de AWS CLI Comandos.

delete-annotation-store-versions

O código de exemplo a seguir mostra como usar `delete-annotation-store-versions`.

AWS CLI

Para excluir uma versão do armazenamento de anotações

O `delete-annotation-store-versions` exemplo a seguir exclui uma versão do armazenamento de anotações.

```
aws omics delete-annotation-store-versions \  
  --name my_annotation_store \  
  --versions my_version
```

Saída:

```
{  
  "errors": []  
}
```

Para obter mais informações, consulte [Criação de novas versões de repositórios de anotações](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [DeleteAnnotationStoreVersions](#) na Referência de AWS CLI Comandos.

delete-annotation-store

O código de exemplo a seguir mostra como usar `delete-annotation-store`.

AWS CLI

Para excluir um repositório de anotações

O `delete-annotation-store` exemplo a seguir exclui um repositório de anotações chamado `my_vcf_store`

```
aws omics delete-annotation-store \  
  --name my_vcf_store
```

Saída:

```
{  
  "status": "DELETING"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteAnnotationStore](#) na Referência de AWS CLI Comandos.

delete-reference-store

O código de exemplo a seguir mostra como usar `delete-reference-store`.

AWS CLI

Para excluir uma loja de referência

O `delete-reference-store` exemplo a seguir exclui um repositório de referência com ID1234567890.

```
aws omics delete-reference-store \  
  --id 1234567890
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteReferenceStore](#) na Referência de AWS CLI Comandos.

delete-reference

O código de exemplo a seguir mostra como usar `delete-reference`.

AWS CLI

Para excluir uma referência

O `delete-reference` exemplo a seguir exclui uma referência.

```
aws omics delete-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteReference](#) na Referência de AWS CLI Comandos.

delete-run-group

O código de exemplo a seguir mostra como usar `delete-run-group`.

AWS CLI

Para excluir um grupo de execução

O `delete-run-group` exemplo a seguir exclui um grupo de execução com ID1234567.

```
aws omics delete-run-group \  
  --id 1234567
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteRunGroup](#) na Referência de AWS CLI Comandos.

`delete-run`

O código de exemplo a seguir mostra como usar `delete-run`.

AWS CLI

Para excluir uma execução de fluxo de trabalho

O `delete-run` exemplo a seguir exclui uma execução com ID1234567.

```
aws omics delete-run \  
  --id 1234567
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteRun](#) na Referência de AWS CLI Comandos.

`delete-sequence-store`

O código de exemplo a seguir mostra como usar `delete-sequence-store`.

AWS CLI

Para excluir um armazenamento de sequências

O `delete-sequence-store` exemplo a seguir exclui um armazenamento de sequência com ID1234567890.

```
aws omics delete-sequence-store \  
  --id 1234567890
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteSequenceStore](#) na Referência de AWS CLI Comandos.

delete-share

O código de exemplo a seguir mostra como usar delete-share.

AWS CLI

Para excluir um compartilhamento de dados de HealthOmics análise

O delete-share exemplo a seguir exclui um compartilhamento de dados analíticos entre contas.

```
aws omics delete-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Saída:

```
{  
  "status": "DELETING"  
}
```

Para obter mais informações, consulte [Compartilhamento entre contas](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [DeleteShare](#) na Referência de AWS CLI Comandos.

delete-variant-store

O código de exemplo a seguir mostra como usar delete-variant-store.

AWS CLI

Para excluir um armazenamento de variantes

O delete-variant-store exemplo a seguir exclui um repositório de variantes chamado my_var_store.


```
aws omics delete-variant-store \  
  --name my_var_store
```

Saída:

```
{  
  "status": "DELETING"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteVariantStore](#) na Referência de AWS CLI Comandos.

delete-workflow

O código de exemplo a seguir mostra como usar delete-workflow.

AWS CLI

Para excluir um fluxo de trabalho

O delete-workflow exemplo a seguir exclui um fluxo de trabalho com ID1234567.

```
aws omics delete-workflow \  
  --id 1234567
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [DeleteWorkflow](#) na Referência de AWS CLI Comandos.

get-annotation-import-job

O código de exemplo a seguir mostra como usar get-annotation-import-job.

AWS CLI

Para visualizar um trabalho de importação de anotações

O get-annotation-import-job exemplo a seguir obtém detalhes sobre um trabalho de importação de anotações.

```
aws omics get-annotation-import-job \  
--job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetAnnotationImportJob](#) na Referência de AWS CLI Comandos.

get-annotation-store-version

O código de exemplo a seguir mostra como usar `get-annotation-store-version`.

AWS CLI

Para recuperar os metadados de uma versão do armazenamento de anotações

O `get-annotation-store-version` exemplo a seguir recupera os metadados da versão solicitada do armazenamento de anotações.

```
aws omics get-annotation-store-version \  
--name my_annotation_store \  
--version-name my_version
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Criação de novas versões de repositórios de anotações](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [GetAnnotationStoreVersion](#) na Referência de AWS CLI Comandos.

get-annotation-store

O código de exemplo a seguir mostra como usar `get-annotation-store`.

AWS CLI

Para ver um repositório de anotações

O `get-annotation-store` exemplo a seguir obtém detalhes sobre um repositório de anotações chamado `my_ann_store`

```
aws omics get-annotation-store \
  --name my_ann_store
```

Saída:

```
{
  "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": {}
}

```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetAnnotationStore](#) na Referência de AWS CLI Comandos.

get-read-set-activation-job

O código de exemplo a seguir mostra como usar `get-read-set-activation-job`.

AWS CLI

Para visualizar um trabalho de ativação do conjunto de leitura

O `get-read-set-activation-job` exemplo a seguir mostra detalhes sobre um trabalho de ativação do conjunto de leitura.

```

aws omics get-read-set-activation-job \
  --sequence-store-id 1234567890 \
  --id 1234567890

```

Saída:

```

{
  "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."  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReadSetActivationJob](#) na Referência de AWS CLI Comandos.

get-read-set-export-job

O código de exemplo a seguir mostra como usar `get-read-set-export-job`.

AWS CLI

Para visualizar um trabalho de exportação do conjunto de leitura

O `get-read-set-export-job` exemplo a seguir obtém detalhes sobre um trabalho de exportação do conjunto de leitura.

```
aws omics get-read-set-export-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Saída:

```
{  
  "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."  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReadSetExportJob](#) na Referência de AWS CLI Comandos.

get-read-set-import-job

O código de exemplo a seguir mostra como usar `get-read-set-import-job`.

AWS CLI

Para visualizar um trabalho de importação de conjunto de leitura

O `get-read-set-import-job` exemplo a seguir obtém detalhes sobre um trabalho de importação de conjunto de leitura.

```
aws omics get-read-set-import-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Saída:

```
{  
  "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."
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReadSetImportJob](#) na Referência de AWS CLI Comandos.

get-read-set-metadata

O código de exemplo a seguir mostra como usar `get-read-set-metadata`.

AWS CLI

Para ver um conjunto de leitura

O `get-read-set-metadata` exemplo a seguir obtém detalhes sobre os arquivos de um conjunto de leitura.

```
aws omics get-read-set-metadata \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReadSetMetadata](#) na Referência de AWS CLI Comandos.

get-read-set

O código de exemplo a seguir mostra como usar `get-read-set`.

AWS CLI

Para baixar um conjunto de leitura

O `get-read-set` exemplo a seguir baixa a parte 3 de um conjunto de leitura como `1234567890.3.bam`.

```
aws omics get-read-set \
  --sequence-store-id 1234567890 \
  --id 1234567890 \
  --part-number 3 1234567890.3.bam
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReadSet](#) na Referência de AWS CLI Comandos.

get-reference-import-job

O código de exemplo a seguir mostra como usar `get-reference-import-job`.

AWS CLI

Para visualizar um trabalho de importação de referência

O `get-reference-import-job` exemplo a seguir obtém detalhes sobre um trabalho de importação de referência.

```
aws omics get-reference-import-job \
  --reference-store-id 1234567890 \
  --id 1234567890
```

Saída:

```
{
  "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."
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReferenceImportJob](#) na Referência de AWS CLI Comandos.

get-reference-metadata

O código de exemplo a seguir mostra como usar `get-reference-metadata`.

AWS CLI

Para ver uma referência

O `get-reference-metadata` exemplo a seguir obtém detalhes sobre uma referência.

```
aws omics get-reference-metadata \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReferenceMetadata](#) na Referência de AWS CLI Comandos.

get-reference-store

O código de exemplo a seguir mostra como usar `get-reference-store`.

AWS CLI

Para ver uma loja de referência

O `get-reference-store` exemplo a seguir mostra detalhes sobre uma loja de referência.

```
aws omics get-reference-store \  
  --id 1234567890
```

Saída:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",  
  "creationTime": "2022-09-23T23:27:20.364Z",  
  "id": "1234567890",  
  "name": "my-rstore-0"  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReferenceStore](#) na Referência de AWS CLI Comandos.

get-reference

O código de exemplo a seguir mostra como usar `get-reference`.

AWS CLI

Para baixar uma referência do genoma

O `get-reference` exemplo a seguir baixa a parte 1 de um genoma como `hg38.1.fa`.

```
aws omics get-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890 \  
  --part-number 1 hg38.1.fa
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetReference](#) na Referência de AWS CLI Comandos.

get-run-group

O código de exemplo a seguir mostra como usar `get-run-group`.

AWS CLI

Para ver um grupo de corrida

O `get-run-group` exemplo a seguir obtém detalhes sobre um grupo de execução.

```
aws omics get-run-group \  
  --id 1234567
```

Saída:

```
{  
  "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": {}  
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetRunGroup](#) na Referência de AWS CLI Comandos.

`get-run-task`

O código de exemplo a seguir mostra como usar `get-run-task`.

AWS CLI

Para ver uma tarefa

O `get-run-task` exemplo a seguir obtém detalhes sobre uma tarefa de fluxo de trabalho.

```
aws omics get-run-task \  
  --id 1234567 \  
  --task-id 1234567
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetRunTask](#) na Referência de AWS CLI Comandos.

get-run

O código de exemplo a seguir mostra como usar `get-run`.

AWS CLI

Para visualizar a execução de um fluxo de trabalho

O `get-run` exemplo a seguir obtém detalhes sobre a execução de um fluxo de trabalho.

```
aws omics get-run \
  --id 1234567
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetRun](#) na Referência de AWS CLI Comandos.

get-sequence-store

O código de exemplo a seguir mostra como usar `get-sequence-store`.

AWS CLI

Para visualizar um armazenamento de sequências

O `get-sequence-store` exemplo a seguir obtém detalhes sobre um armazenamento de sequência com ID1234567890.

```
aws omics get-sequence-store \  
  --id 1234567890
```

Saída:

```
{  
  "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",  
  "creationTime": "2022-11-23T19:55:48.376Z",  
  "id": "1234567890",  
  "name": "my-seq-store"  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetSequenceStore](#) na Referência de AWS CLI Comandos.

get-share

O código de exemplo a seguir mostra como usar `get-share`.

AWS CLI

Para recuperar os metadados sobre um compartilhamento de dados analíticos HealthOmics

O `get-share` exemplo a seguir recupera os metadados de um compartilhamento de dados analíticos entre contas.

```
aws omics get-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Compartilhamento entre contas](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [GetShare](#) na Referência de AWS CLI Comandos.

get-variant-import-job

O código de exemplo a seguir mostra como usar `get-variant-import-job`.

AWS CLI

Para visualizar um trabalho de importação de variantes

O `get-variant-import-job` exemplo a seguir obtém detalhes sobre um trabalho de importação de variantes.

```

aws omics get-variant-import-job \
  --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetVariantImportJob](#) na Referência de AWS CLI Comandos.

get-variant-store

O código de exemplo a seguir mostra como usar `get-variant-store`.

AWS CLI

Para ver uma loja de variantes

O `get-variant-store` exemplo a seguir mostra detalhes sobre uma loja de variantes.

```
aws omics get-variant-store \  
  --name my_var_store
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetVariantStore](#) na Referência de AWS CLI Comandos.

get-workflow

O código de exemplo a seguir mostra como usar `get-workflow`.

AWS CLI

Para visualizar um fluxo de trabalho

O `get-workflow` exemplo a seguir obtém detalhes sobre um fluxo de trabalho com ID1234567.

```
aws omics get-workflow \  
  --id 1234567
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [GetWorkflow](#) na Referência de AWS CLI Comandos.

list-annotation-import-jobs

O código de exemplo a seguir mostra como usar `list-annotation-import-jobs`.

AWS CLI

Para obter uma lista de trabalhos de importação de anotações

Veja a seguir `list-annotation-import-jobs` uma lista de trabalhos de importação de anotações.

```
aws omics list-annotation-import-jobs
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListAnnotationImportJobs](#) na Referência de AWS CLI Comandos.

list-annotation-store-versions

O código de exemplo a seguir mostra como usar `list-annotation-store-versions`.

AWS CLI

Para listar todas as versões de um repositório de anotações.

O `list-annotation-store-versions` exemplo a seguir lista todas as versões existentes de um repositório de anotações.

```

aws omics list-annotation-store-versions \
  --name my_annotation_store

```

Saída:

```

{
  "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
  }
}
```

Para obter mais informações, consulte [Criação de novas versões de repositórios de anotações](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [ListAnnotationStoreVersions](#) na Referência de AWS CLI Comandos.

list-annotation-stores

O código de exemplo a seguir mostra como usar `list-annotation-stores`.

AWS CLI

Para obter uma lista de lojas de anotações

O `list-annotation-stores` exemplo a seguir obtém uma lista de lojas de anotações.

```
aws omics list-annotation-stores
```

Saída:

```
{
  "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"
    }
]
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListAnnotationStores](#) na Referência de AWS CLI Comandos.

list-multipart-read-set-uploads

O código de exemplo a seguir mostra como usar `list-multipart-read-set-uploads`.

AWS CLI

Para listar todas as leituras em várias partes, configure os uploads e seus status.

O `list-multipart-read-set-uploads` exemplo a seguir lista todos os uploads de conjuntos de leitura de várias partes e seus status.

```
aws omics list-multipart-read-set-uploads \
  --sequence-store-id 0123456789
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Carregamento direto para um armazenamento de sequências](#) no Guia AWS HealthOmics do usuário.

- Para API obter detalhes, consulte [ListMultipartReadSetUploads](#) na Referência de AWS CLI Comandos.

list-read-set-activation-jobs

O código de exemplo a seguir mostra como usar `list-read-set-activation-jobs`.

AWS CLI

Para obter uma lista de trabalhos de ativação do conjunto de leitura

O `list-read-set-activation-jobs` exemplo a seguir obtém uma lista de trabalhos de ativação para um armazenamento de sequência com id1234567890.

```
aws omics list-read-set-activation-jobs \  
--sequence-store-id 1234567890
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReadSetActivationJobs](#) na Referência de AWS CLI Comandos.

list-read-set-export-jobs

O código de exemplo a seguir mostra como usar `list-read-set-export-jobs`.

AWS CLI

Para obter uma lista de trabalhos de exportação do conjunto de leitura

O `list-read-set-export-jobs` exemplo a seguir obtém uma lista de trabalhos de exportação para um armazenamento de sequência com id1234567890.

```
aws omics list-read-set-export-jobs \  
--sequence-store-id 1234567890
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReadSetExportJobs](#) na Referência de AWS CLI Comandos.

list-read-set-import-jobs

O código de exemplo a seguir mostra como usar `list-read-set-import-jobs`.

AWS CLI

Para obter uma lista de trabalhos de importação de conjuntos de leitura

O `list-read-set-import-jobs` exemplo a seguir obtém uma lista de trabalhos de importação para um armazenamento de sequência com id1234567890.

```
aws omics list-read-set-import-jobs \  

```

```
--sequence-store-id 1234567890
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReadSetImportJobs](#) na Referência de AWS CLI Comandos.

list-read-set-upload-parts

O código de exemplo a seguir mostra como usar `list-read-set-upload-parts`.

AWS CLI

Para listar todas as partes em um upload de várias partes solicitado para um armazenamento de sequências.

O `list-read-set-upload-parts` exemplo a seguir lista todas as partes em um upload de várias partes solicitado para um armazenamento de sequências.

```
aws omics list-read-set-upload-parts \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --part-source SOURCE1
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Carregamento direto para um armazenamento de sequências](#) no Guia AWS HealthOmics do usuário.

- Para API obter detalhes, consulte [ListReadSetUploadParts](#) na Referência de AWS CLI Comandos.

list-read-sets

O código de exemplo a seguir mostra como usar `list-read-sets`.

AWS CLI

Para obter uma lista de conjuntos de leitura

O `list-read-sets` exemplo a seguir obtém uma lista de conjuntos de leitura para um armazenamento de sequência com id1234567890.

```
aws omics list-read-sets \  
--sequence-store-id 1234567890
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReadSets](#) na Referência de AWS CLI Comandos.

list-reference-import-jobs

O código de exemplo a seguir mostra como usar `list-reference-import-jobs`.

AWS CLI

Para obter uma lista de trabalhos de importação de referência

O `list-reference-import-jobs` exemplo a seguir obtém uma lista de trabalhos de importação de referência para uma loja de referência com id1234567890.

```
aws omics list-reference-import-jobs \  
--reference-store-id 1234567890
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReferenceImportJobs](#) na Referência de AWS CLI Comandos.

list-reference-stores

O código de exemplo a seguir mostra como usar `list-reference-stores`.

AWS CLI

Para obter uma lista de lojas de referência

O `list-reference-stores` exemplo a seguir obtém uma lista de lojas de referência.

```
aws omics list-reference-stores
```

Saída:

```
{
  "referenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
      "creationTime": "2022-11-22T22:13:25.947Z",
      "id": "1234567890",
      "name": "my-ref-store"
    }
  ]
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReferenceStores](#) na Referência de AWS CLI Comandos.

list-references

O código de exemplo a seguir mostra como usar `list-references`.

AWS CLI

Para obter uma lista de referências

O `list-references` exemplo a seguir obtém uma lista de referências de genoma para um repositório de referência com id1234567890.

```
aws omics list-references \
  --reference-store-id 1234567890
```

Saída:

```
{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListReferences](#) na Referência de AWS CLI Comandos.

list-run-groups

O código de exemplo a seguir mostra como usar `list-run-groups`.

AWS CLI

Para obter uma lista de grupos de corrida

O `list-run-groups` exemplo a seguir obtém uma lista de grupos de execução.

```
aws omics list-run-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```


Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListRunGroups](#) na Referência de AWS CLI Comandos.

list-run-tasks

O código de exemplo a seguir mostra como usar `list-run-tasks`.

AWS CLI

Para obter uma lista de tarefas

O `list-run-tasks` exemplo a seguir obtém uma lista de tarefas para a execução de um fluxo de trabalho.

```
aws omics list-run-tasks \  
  --id 1234567
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListRunTasks](#) na Referência de AWS CLI Comandos.

list-runs

O código de exemplo a seguir mostra como usar `list-runs`.

AWS CLI

Para obter uma lista das execuções do fluxo de trabalho

O `list-runs` exemplo a seguir obtém uma lista das execuções do fluxo de trabalho.

```
aws omics list-runs
```

Saída:

```
{
  "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"
    }
]
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListRuns](#) na Referência de AWS CLI Comandos.

list-sequence-stores

O código de exemplo a seguir mostra como usar `list-sequence-stores`.

AWS CLI

Para obter uma lista de armazenamentos de sequências

O `list-sequence-stores` exemplo a seguir obtém uma lista de armazenamentos de sequências.

```
aws omics list-sequence-stores
```

Saída:

```
{
  "sequenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
      "creationTime": "2022-11-23T01:24:33.629Z",
```

```
        "id": "1234567890",
        "name": "my-seq-store"
      }
    ]
  }
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListSequenceStores](#) na Referência de AWS CLI Comandos.

list-shares

O código de exemplo a seguir mostra como usar `list-shares`.

AWS CLI

Para listar os compartilhamentos disponíveis de dados de HealthOmics análise

O `list-shares` exemplo a seguir lista todos os compartilhamentos que foram criados para um proprietário do recurso.

```
aws omics list-shares \
  --resource-owner SELF
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Compartilhamento entre contas](#) no Guia do AWS HealthOmics usuário.

- Para API obter detalhes, consulte [ListShares](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para obter uma lista de tags

O `list-tags-for-resource` exemplo a seguir obtém uma lista de tags para um fluxo de trabalho com `id1234567`.

```

aws omics list-tags-for-resource \
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567

```

Saída:

```

{
  "tags": {
    "department": "analytics"
  }
}

```

Para obter mais informações, consulte [Marcar recursos no Amazon Omics no Amazon Omics Developer Guide](#).

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-variant-import-jobs

O código de exemplo a seguir mostra como usar `list-variant-import-jobs`.

AWS CLI

Para obter uma lista de trabalhos de importação de variantes

O `list-variant-import-jobs` exemplo a seguir obtém uma lista de trabalhos de importação de variantes.

```
aws omics list-variant-import-jobs
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListVariantImportJobs](#) na Referência de AWS CLI Comandos.

list-variant-stores

O código de exemplo a seguir mostra como usar `list-variant-stores`.

AWS CLI

Para obter uma lista de lojas variantes

O `list-variant-stores` exemplo a seguir obtém uma lista de lojas de variantes.

```
aws omics list-variant-stores
```

Saída:

```
{
  "variantStores": [
    {
      "creationTime": "2022-11-23T22:09:07.534499Z",
      "id": "02dexplcfdd",
      "name": "my_var_store",
      "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
      },
      "status": "CREATING",
      "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
my_var_store",
      "storeSizeBytes": 0,
      "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"  
    }  
]  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListVariantStores](#) na Referência de AWS CLI Comandos.

list-workflows

O código de exemplo a seguir mostra como usar `list-workflows`.

AWS CLI

Para obter uma lista de fluxos de trabalho

O `list-workflows` exemplo a seguir obtém uma lista de fluxos de trabalho.

```
aws omics list-workflows
```

Saída:

```
{  
  "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"
      }
    ]
  }
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [ListWorkflows](#) na Referência de AWS CLI Comandos.

start-annotation-import-job

O código de exemplo a seguir mostra como usar `start-annotation-import-job`.

AWS CLI

Para importar anotações

O `start-annotation-import-job` exemplo a seguir importa anotações do 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
```

Saída:

```
{
  "jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [StartAnnotationImportJob](#) na Referência de AWS CLI Comandos.

start-read-set-activation-job

O código de exemplo a seguir mostra como usar `start-read-set-activation-job`.


```
--destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

Saída:

```
{
  "creationTime": "2022-12-06T22:37:18.612Z",
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [StartReadSetExportJob](#) na Referência de AWS CLI Comandos.

start-read-set-import-job

O código de exemplo a seguir mostra como usar `start-read-set-import-job`.

AWS CLI

Para importar um conjunto de leitura

O `start-read-set-import-job` exemplo a seguir importa um conjunto de leitura.

```
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` é um documento com o seguinte conteúdo. 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"
  }
]
```

Saída:

```
{
  "creationTime": "2022-11-23T01:36:38.158Z",
  "id": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [StartReadSetImportJob](#) na Referência de AWS CLI Comandos.

start-reference-import-job

O código de exemplo a seguir mostra como usar `start-reference-import-job`.

AWS CLI

Para importar um genoma de referência

O `start-reference-import-job` exemplo a seguir importa um genoma de referência do 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
```

Saída:

```
{
  "creationTime": "2022-11-22T22:25:41.124Z",
  "id": "1234567890",
  "referenceStoreId": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "status": "SUBMITTED"
}
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [StartReferenceImportJob](#) na Referência de AWS CLI Comandos.

start-run

O código de exemplo a seguir mostra como usar `start-run`.

AWS CLI

Para executar um fluxo de trabalho

O `start-run` exemplo a seguir executa um fluxo de trabalho com 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` é um documento com o seguinte conteúdo. 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"
}

```

Saída:

```

{
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
  "id": "1234567",
  "status": "PENDING",
  "tags": {}
}

```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

Para carregar arquivos de origem do Amazon Omics

Você também pode carregar arquivos de origem do armazenamento do Amazon Omics usando serviços URIs específicos. O exemplo de arquivo workflow-inputs.json a seguir usa o Amazon URIs Omics para fontes de conjuntos de leitura e de referência do genoma.

```

{
  "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"
}

```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [StartRun](#) na Referência de AWS CLI Comandos.

start-variant-import-job

O código de exemplo a seguir mostra como usar `start-variant-import-job`.

AWS CLI

Para importar um arquivo de variante

O `start-variant-import-job` exemplo a seguir importa um arquivo de variante de 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
```

Saída:

```
{  
  "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [StartVariantImportJob](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir adiciona uma `department` tag a um fluxo de trabalho com `id1234567`.

```
aws omics tag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tags department=analytics
```

Para obter mais informações, consulte [Marcar recursos no Amazon Omics no Amazon Omics Developer Guide](#).

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove a `department` tag de um fluxo de trabalho.

```
aws omics untag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tag-keys department
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-annotation-store

O código de exemplo a seguir mostra como usar `update-annotation-store`.

AWS CLI

Para atualizar um repositório de anotações

O `update-annotation-store` exemplo a seguir atualiza a descrição de um repositório de anotações chamado `my_vcf_store`

```
aws omics update-annotation-store \  
  --name my_vcf_store \  
  --description my_vcf_store
```



```
--description "VCF annotation store"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [UpdateAnnotationStore](#) na Referência de AWS CLI Comandos.

update-run-group

O código de exemplo a seguir mostra como usar `update-run-group`.

AWS CLI

Para atualizar um grupo de execução

O `update-run-group` exemplo a seguir atualiza as configurações de um grupo de execução com id1234567.

```
aws omics update-run-group \
  --id 1234567 \
  --max-cpus 10
```

Saída:

```
{
```

```
"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": {}
}
```

Para obter mais informações, consulte [Fluxos de trabalho do Omics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [UpdateRunGroup](#) na Referência de AWS CLI Comandos.

update-variant-store

O código de exemplo a seguir mostra como usar `update-variant-store`.

AWS CLI

Para atualizar uma loja de variantes

O `update-variant-store` exemplo a seguir atualiza a descrição de um repositório de variantes chamado `my_var_store`.

```
aws omics update-variant-store \
  --name my_var_store \
  --description "variant store"
```

Saída:

```
{
  "creationTime": "2022-11-23T22:09:07.534499Z",
  "description": "variant store",
  "id": "02dexplcfdd",
  "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"  
}
```

Para obter mais informações, consulte [Omics Analytics](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [UpdateVariantStore](#) na Referência de AWS CLI Comandos.

update-workflow

O código de exemplo a seguir mostra como usar `update-workflow`.

AWS CLI

Para atualizar um fluxo de trabalho

O `update-workflow` exemplo a seguir atualiza a descrição de um fluxo de trabalho com ID1234567.

```
aws omics update-workflow \  
  --id 1234567 \  
  --description "copy workflow"
```

Para obter mais informações, consulte [Omics Storage](#) no Amazon Omics Developer Guide.

- Para API obter detalhes, consulte [UpdateWorkflow](#) na Referência de AWS CLI Comandos.

upload-read-set-part

O código de exemplo a seguir mostra como usar `upload-read-set-part`.

AWS CLI

Para fazer o upload de uma parte do conjunto de leitura.

O `upload-read-set-part` exemplo a seguir carrega uma parte específica de um conjunto de leitura.

```
aws omics upload-read-set-part \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --part-source SOURCE1 \  
  --part-number 1
```

```
--part-number 1 \  
--payload /path/to/file/read_1_part_1.fastq.gz
```

Saída:

```
{  
  "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"  
}
```

Para obter mais informações, consulte [Carregamento direto para um armazenamento de sequências](#) no Guia AWS HealthOmics do usuário.

- Para API obter detalhes, consulte [UploadReadSetPart](#)na Referência de AWS CLI Comandos.

IAMexemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface withIAM.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-client-id-to-open-id-connect-provider

O código de exemplo a seguir mostra como usar `add-client-id-to-open-id-connect-provider`.

AWS CLI

Para adicionar um ID de cliente (público) a um provedor Open-ID Connect (OIDC)

O `add-client-id-to-open-id-connect-provider` comando a seguir adiciona o ID do cliente `my-application-ID` ao OIDC provedor `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
```

Este comando não produz saída.

Para criar um OIDC provedor, use o `create-open-id-connect-provider` comando.

Para obter mais informações, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [AddClientIdToOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

add-role-to-instance-profile

O código de exemplo a seguir mostra como usar `add-role-to-instance-profile`.

AWS CLI

Adicionar um perfil a um perfil de instância

O comando `add-role-to-instance-profile` a seguir adiciona o perfil denominado `S3Access` ao perfil de instância denominado `Webserver`.

```
aws iam add-role-to-instance-profile \  
  --role-name S3Access \  
  --instance-profile-name Webserver
```

Este comando não produz saída.

Para criar um perfil de instância, use o comando `create-instance-profile`.

Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [AddRoleToInstanceProfile](#) na Referência de AWS CLI Comandos.

add-user-to-group

O código de exemplo a seguir mostra como usar `add-user-to-group`.

AWS CLI

Para adicionar um usuário a um IAM grupo

O `add-user-to-group` comando a seguir adiciona um IAM usuário chamado Bob ao IAM grupo chamado Admins.

```
aws iam add-user-to-group \  
  --user-name Bob \  
  --group-name Admins
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar e remover usuários em um IAM grupo](#) de usuários no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [AddUserToGroup](#) na Referência de AWS CLI Comandos.

attach-group-policy

O código de exemplo a seguir mostra como usar `attach-group-policy`.

AWS CLI

Para anexar uma política gerenciada a um IAM grupo

O `attach-group-policy` comando a seguir anexa a política AWS gerenciada nomeada `ReadOnlyAccess` ao IAM grupo nomeado `Finance`.

```
aws iam attach-group-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --group-name Finance
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas gerenciadas e políticas em linha](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [AttachGroupPolicy](#) na Referência de AWS CLI Comandos.

attach-role-policy

O código de exemplo a seguir mostra como usar `attach-role-policy`.

AWS CLI

Para anexar uma política gerenciada a uma IAM função

O `attach-role-policy` comando a seguir anexa a política AWS gerenciada nomeada `ReadOnlyAccess` à IAM função nomeada `ReadOnlyRole`.

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --role-name ReadOnlyRole
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas gerenciadas e políticas em linha](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [AttachRolePolicy](#) na Referência de AWS CLI Comandos.

attach-user-policy

O código de exemplo a seguir mostra como usar `attach-user-policy`.

AWS CLI

Para anexar uma política gerenciada a um IAM usuário

O `attach-user-policy` comando a seguir anexa a política AWS gerenciada nomeada `AdministratorAccess` ao IAM usuário nomeado `Alice`.

```
aws iam attach-user-policy \  
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \  
  --user-name Alice
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas gerenciadas e políticas em linha](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [AttachUserPolicy](#) na Referência de AWS CLI Comandos.

change-password

O código de exemplo a seguir mostra como usar `change-password`.

AWS CLI

Para alterar a senha do seu IAM usuário

Para alterar a senha IAM do seu usuário, recomendamos usar o `--cli-input-json` parâmetro para transmitir um JSON arquivo que contenha suas senhas antigas e novas. Com esse método, você pode usar senhas fortes com caracteres não alfanuméricos. Pode ser difícil usar senhas com caracteres não alfanuméricos quando elas são transmitidas como parâmetros da linha de comando. Para usar o parâmetro `--cli-input-json`, comece usando o comando `change-password` com o parâmetro `--generate-cli-skeleton`, como no exemplo a seguir.

```
aws iam change-password \  
  --generate-cli-skeleton > change-password.json
```

O comando anterior cria um JSON arquivo chamado `change-password.json` que você pode usar para preencher suas senhas antigas e novas. Por exemplo, o perfil pode ter a aparência a seguir.

```
{  
  "OldPassword": "3s0K_;xh4~8XXI",  
  "NewPassword": "]35d/{pB9Fo9wJ"  
}
```

Em seguida, para alterar sua senha, use o `change-password` comando novamente, desta vez passando o `--cli-input-json` parâmetro para especificar seu JSON arquivo. O `change-password` comando a seguir usa o `--cli-input-json` parâmetro com um JSON arquivo chamado `change-password.json`.

```
aws iam change-password \  
  --cli-input-json file://change-password.json
```

Este comando não produz saída.

Esse comando só pode ser chamado por IAM usuários. Se esse comando for chamado usando as credenciais AWS da conta (raiz), o comando retornará um `InvalidUserType` erro.

Para obter mais informações, consulte [Como um IAM usuário altera sua própria senha](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ChangePassword](#) na Referência de AWS CLI Comandos.

create-access-key

O código de exemplo a seguir mostra como usar `create-access-key`.

AWS CLI

Para criar uma chave de acesso para um IAM usuário

O `create-access-key` comando a seguir cria uma chave de acesso (ID da chave de acesso e chave de acesso secreta) para o IAM usuário nomeado `Bob`.

```
aws iam create-access-key \  
  --user-name Bob
```

Saída:

```
{  
  "AccessKey": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CreateDate": "2015-03-09T18:39:23.411Z",  
    "SecretAccessKey": "wJa1rXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"  
  }  
}
```

Armazene a chave de acesso secreta em um local seguro. Se ela for perdida, não será possível recuperá-la e você deverá criar uma nova chave de acesso.

Para obter mais informações, consulte [Gerenciamento de chaves de acesso para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [CreateAccessKey](#) na Referência de AWS CLI Comandos.

create-account-alias

O código de exemplo a seguir mostra como usar `create-account-alias`.

AWS CLI

Como criar um alias da conta

O `create-account-alias` comando a seguir cria o alias `examplecorp` para sua AWS conta.

```
aws iam create-account-alias \  
  --account-alias examplecorp
```

Este comando não produz saída.

Para obter mais informações, consulte [o ID AWS da sua conta e seu alias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreateAccountAlias](#) na Referência de AWS CLI Comandos.

create-group

O código de exemplo a seguir mostra como usar `create-group`.

AWS CLI

Para criar um IAM grupo

O `create-group` comando a seguir cria um IAM grupo chamado `Admins`.

```
aws iam create-group \  
  --group-name Admins
```

Saída:

```
{  
  "Group": {  
    "Path": "/",  
    "CreateDate": "2015-03-09T20:30:24.940Z",  
    "GroupId": "AIDGPMS9R04H3FEXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:group/Admins",  
    "GroupName": "Admins"
```

```
}  
}
```

Para obter mais informações, consulte [Criação IAM de grupos](#) de AWS IAM usuários no Guia do usuário.

- Para API obter detalhes, consulte [CreateGroup](#) na Referência de AWS CLI Comandos.

create-instance-profile

O código de exemplo a seguir mostra como usar `create-instance-profile`.

AWS CLI

Como criar um perfil de instância

O comando `create-instance-profile`, apresentado a seguir, cria um perfil de instância denominado `Webserver`.

```
aws iam create-instance-profile \  
--instance-profile-name Webserver
```

Saída:

```
{  
  "InstanceProfile": {  
    "InstanceId": "AIPAJMBYC7DLSPEXAMPLE",  
    "Roles": [],  
    "CreateDate": "2015-03-09T20:33:19.626Z",  
    "InstanceProfileName": "Webserver",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"  
  }  
}
```

Para adicionar um perfil a um perfil de instância, use o comando `add-role-to-instance-profile`.

Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [CreateInstanceProfile](#) na Referência de AWS CLI Comandos.

create-login-profile

O código de exemplo a seguir mostra como usar `create-login-profile`.

AWS CLI

Para criar uma senha para um IAM usuário

Para criar uma senha para um IAM usuário, recomendamos usar o `--cli-input-json` parâmetro para passar um JSON arquivo que contenha a senha. Usando esse método, você pode criar uma senha forte com caracteres não alfanuméricos. Pode ser difícil criar uma senha com caracteres não alfanuméricos ao transmiti-la como parâmetro da linha de comando.

Para usar o parâmetro `--cli-input-json`, comece usando o comando `create-login-profile` com o parâmetro `--generate-cli-skeleton`, como no exemplo a seguir.

```
aws iam create-login-profile \  
  --generate-cli-skeleton > create-login-profile.json
```

O comando anterior cria um JSON arquivo chamado `create-login-profile.json` que você pode usar para preencher as informações de um comando subsequente `create-login-profile`. Por exemplo:

```
{  
  "UserName": "Bob",  
  "Password": "&1-3a6u:RA@djs",  
  "PasswordResetRequired": true  
}
```

Em seguida, para criar uma senha para um IAM usuário, use o `create-login-profile` comando novamente, desta vez passando o `--cli-input-json` parâmetro para especificar seu JSON arquivo. O `create-login-profile` comando a seguir usa o `--cli-input-json` parâmetro com um JSON arquivo chamado `create-login-profile.json`.

```
aws iam create-login-profile \  
  --cli-input-json file://create-login-profile.json
```

Saída:

```
{
```

```
"LoginProfile": {
  "UserName": "Bob",
  "CreateDate": "2015-03-10T20:55:40.274Z",
  "PasswordResetRequired": true
}
```

Se a nova senha violar a política de senha da conta, o comando retornará um erro `PasswordPolicyViolation`.

Para alterar a senha de um usuário que já tem uma, use `update-login-profile`. Para definir uma política de senha da conta, use o comando `update-account-password-policy`.

Se a política de senha da conta permitir, IAM os usuários poderão alterar suas próprias senhas usando o `change-password` comando.

Para obter mais informações, consulte [Gerenciamento de senhas para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [CreateLoginProfile](#) na Referência de AWS CLI Comandos.

create-open-id-connect-provider

O código de exemplo a seguir mostra como usar `create-open-id-connect-provider`.

AWS CLI

Para criar um provedor OpenID Connect (OIDC)

Para criar um provedor OpenID Connect (OIDC), recomendamos usar o `--cli-input-json` parâmetro para transmitir um JSON arquivo que contenha os parâmetros necessários. Ao criar um OIDC provedor, você deve passar o URL do provedor e começar com `https://`. URL Pode ser difícil passar o URL como parâmetro de linha de comando, porque os caracteres de dois pontos (`:`) e barra (`/`) têm um significado especial em alguns ambientes de linha de comando. Usar o parâmetro `--cli-input-json` contorna essa limitação.

Para usar o parâmetro `--cli-input-json`, comece usando o comando `create-open-id-connect-provider` com o parâmetro `--generate-cli-skeleton`, como no exemplo a seguir.

```
aws iam create-open-id-connect-provider \
```

```
--generate-cli-skeleton > create-open-id-connect-provider.json
```

O comando anterior cria um JSON arquivo chamado `create-open-id-connect-provider.json` que você pode usar para preencher as informações de um comando subsequente. `create-open-id-connect-provider` Por exemplo:

```
{
  "Url": "https://server.example.com",
  "ClientIDList": [
    "example-application-ID"
  ],
  "ThumbprintList": [
    "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
  ]
}
```

Em seguida, para criar o provedor OpenID Connect (OIDC), use o `create-open-id-connect-provider` comando novamente, desta vez passando o `--cli-input-json` parâmetro para especificar seu JSON arquivo. O `create-open-id-connect-provider` comando a seguir usa o `--cli-input-json` parâmetro com um JSON arquivo chamado `create-open-id-connect-provider.json`.

```
aws iam create-open-id-connect-provider \  
--cli-input-json file://create-open-id-connect-provider.json
```

Saída:

```
{
  "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

Para obter mais informações sobre OIDC provedores, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

Para obter mais informações sobre como obter impressões digitais de um OIDC provedor, consulte [Obter a impressão digital de um provedor de identidade OpenID Connect](#) no Guia do usuário.AWS IAM

- Para API obter detalhes, consulte [CreateOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

create-policy-version

O código de exemplo a seguir mostra como usar `create-policy-version`.

AWS CLI

Como criar uma nova versão de uma política gerenciada

Este exemplo cria uma nova v2 versão da IAM política cujo ARN é `arn:aws:iam::123456789012:policy/MyPolicy` e a torna a versão padrão.

```
aws iam create-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --policy-document file://NewPolicyVersion.json \  
  --set-as-default
```

Saída:

```
{  
  "PolicyVersion": {  
    "CreateDate": "2015-06-16T18:56:03.721Z",  
    "VersionId": "v2",  
    "IsDefaultVersion": true  
  }  
}
```

Para obter mais informações, consulte [IAM Políticas de controle de versão](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreatePolicyVersion](#) na Referência de AWS CLI Comandos.

create-policy

O código de exemplo a seguir mostra como usar `create-policy`.

AWS CLI

Exemplo 1: como criar uma política gerenciada pelo cliente

O comando apresentado a seguir cria uma política gerenciada pelo cliente denominada `my-policy`.

```
aws iam create-policy \  
  --policy-name my-policy
```

```
--policy-name my-policy \  
--policy-document file://policy
```

O arquivo `policy` é um JSON documento na pasta atual que concede acesso somente de leitura à `shared` pasta em um bucket do Amazon S3 chamado `my-bucket`

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:Get*",  
        "s3:List*"  
      ],  
      "Resource": [  
        "arn:aws:s3:::my-bucket/shared/*"  
      ]  
    }  
  ]  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações sobre o uso de arquivos como entrada para parâmetros de string, consulte [Especificar valores de parâmetros para o AWS CLI](#) no Guia AWS CLI do usuário.

Exemplo 2: como criar uma política gerenciada pelo cliente com uma descrição

O seguinte comando cria uma política gerenciada pelo cliente denominada `my-policy` com uma descrição imutável:

```
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"
```

O arquivo `policy.json` é um JSON documento na pasta atual que concede acesso a todas as ações Put, List e Get de um bucket do Amazon S3 chamado `my-bucket`

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:ListBucket*",  
        "s3:PutBucket*",  
        "s3:GetBucket*"  
      ],  
      "Resource": [  
        "arn:aws:s3:::my-bucket"  
      ]  
    }  
  ]  
}
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações sobre políticas baseadas em identidade, consulte Políticas baseadas [em identidade e políticas baseadas em recursos](#) no Guia do usuário.AWS IAM

Exemplo 3: como criar uma política gerenciada pelo cliente com etiquetas

O comando apresentado a seguir cria uma política gerenciada pelo cliente, denominada `my-policy`, com etiquetas. Este exemplo usa o sinalizador de `--tags` parâmetro com as seguintes tags JSON formatadas: `'{"Key": "Department", "Value": "Accounting"}'` `'{"Key": "Location", "Value": "Seattle"}'` Como alternativa, o sinalizador `--tags` pode ser usado com etiquetas no formato abreviado: `'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"}'

```

O arquivo `policy.json` é um JSON documento na pasta atual que concede acesso a todas as ações Put, List e Get de um bucket do Amazon S3 chamado. `my-bucket`

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket*",
        "s3:PutBucket*",
        "s3:GetBucket*"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket"
      ]
    }
  ]
}

```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Para obter mais informações sobre políticas de marcação, consulte [Marcação de políticas gerenciadas pelo cliente](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreatePolicy](#) na Referência de AWS CLI Comandos.

create-role

O código de exemplo a seguir mostra como usar `create-role`.

AWS CLI

Exemplo 1: Para criar uma IAM função

O comando `create-role`, apresentado a seguir, cria um perfil, denominado `Test-Role`, e anexa uma política de confiança a ele.

```
aws iam create-role \  
  --role-name Test-Role \  
  --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

Saída:

```
{  
  "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"  
  }  
}
```

A política de confiança é definida como um JSON documento no arquivo Test-role-trust-policy.json. (O nome e a extensão do arquivo não têm significado.) A política de confiança deve especificar uma entidade principal.

Para anexar uma política de permissões a um perfil, use o comando `put-role-policy`.

Para obter mais informações, consulte [Criação de IAM funções](#) no Guia AWS IAM do usuário.

Exemplo 2: Para criar uma IAM função com duração máxima de sessão especificada

O comando `create-role`, apresentado a seguir, cria um perfil denominado `Test-Role` e define a duração máxima da sessão como 7.200 segundos (duas horas).

```
aws iam create-role \  
  --role-name Test-Role \  
  --assume-role-policy-document file://Test-Role-Trust-Policy.json \  
  --max-session-duration 7200
```

Saída:

```
{  
  "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"
    }
  ]
}
}
}
}

```

Para obter mais informações, consulte [Modificar a duração máxima da sessão de uma função \(AWS API\)](#) no Guia do AWS IAM usuário.

Exemplo 3: Para criar uma IAM função com tags

O comando a seguir cria uma IAM função Test-Role com tags. Este exemplo usa o sinalizador de `--tags` parâmetro com as seguintes tags JSON formatadas: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'` Como alternativa, o sinalizador `--tags` pode ser usado com etiquetas no formato abreviado: `'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"}'

```

Saída:

```

{
  "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"
  }
]
}
```

Para obter mais informações, consulte Como [marcar IAM funções](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreateRole](#) na Referência de AWS CLI Comandos.

create-saml-provider

O código de exemplo a seguir mostra como usar `create-saml-provider`.

AWS CLI

Para criar um SAML provedor

Este exemplo cria um novo SAML provedor em IAM named `MySAMLProvider`. Ela é descrita pelo documento de SAML metadados encontrado no arquivo `SAMLMetaData.xml`.

```
aws iam create-saml-provider \  
  --saml-metadata-document file://SAMLMetaData.xml \  
  --name MySAMLProvider
```

Saída:

```
{  
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"  
}
```

Para obter mais informações, consulte [Criação de provedores de IAM SAML identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [C reateSAMLProvider](#) na Referência de AWS CLI Comandos.

create-service-linked-role

O código de exemplo a seguir mostra como usar create-service-linked-role.

AWS CLI

Como criar um perfil vinculado ao serviço

O create-service-linked-role exemplo a seguir cria uma função vinculada ao serviço para o AWS serviço especificado e anexa a descrição especificada.

```
aws iam create-service-linked-role \  
  --aws-service-name lex.amazonaws.com \  
  --description "My service-linked role to support Lex"
```

Saída:

```
{  
  "Role": {  
    "Path": "/aws-service-role/lex.amazonaws.com/",  
    "RoleName": "AWSServiceRoleForLexBots",  
    "RoleId": "ARO1234567890EXAMPLE",  
    "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"
        ]
      }
    }
  ]
}
```

Para obter mais informações, consulte [Usando funções vinculadas a serviços](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreateServiceLinkedRole](#) na Referência de AWS CLI Comandos.

create-service-specific-credential

O código de exemplo a seguir mostra como usar `create-service-specific-credential`.

AWS CLI

Crie um conjunto de credenciais específicas do serviço para um usuário

O `create-service-specific-credential` exemplo a seguir cria um nome de usuário e uma senha que podem ser usados para acessar somente o serviço configurado.

```
aws iam create-service-specific-credential \
  --user-name sofia \
  --service-name codecommit.amazonaws.com
```

Saída:


```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Criar credenciais do Git para HTTPS conexões CodeCommit](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [CreateServiceSpecificCredential](#) na Referência de AWS CLI Comandos.

create-user

O código de exemplo a seguir mostra como usar create-user.

AWS CLI

Exemplo 1: Para criar um IAM usuário

O create-user comando a seguir cria um IAM usuário nomeado Bob na conta atual.

```
aws iam create-user \
  --user-name Bob
```

Saída:

```
{
  "User": {
    "UserName": "Bob",
    "Path": "/",
    "CreateDate": "2023-06-08T03:20:41.270Z",
    "UserId": "AIDAIOSFODNN7EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Bob"
  }
}
```

Para obter mais informações, consulte [Criação de um IAM usuário em sua AWS conta](#) no Guia AWS IAM do usuário.

Exemplo 2: Para criar um IAM usuário em um caminho especificado

O `create-user` comando a seguir cria um IAM usuário nomeado Bob no caminho especificado.

```
aws iam create-user \  
  --user-name Bob \  
  --path /division_abc/subdivision_xyz/
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte os [IAM identificadores](#) no Guia do AWS IAM usuário.

Exemplo 3: Para criar um IAM usuário com tags

O `create-user` comando a seguir cria um IAM usuário chamado Bob com tags. Este exemplo usa o sinalizador de `--tags` parâmetro com as seguintes tags JSON formatadas: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'` Como alternativa, o sinalizador `--tags` pode ser usado com etiquetas no formato abreviado: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-user \  
  --user-name Bob \  
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
  "Value": "Seattle"}'
```

Saída:

```
{
```

```

    "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"
        }
      ]
    }
  }
}

```

Para obter mais informações, consulte Como [marcar IAM usuários](#) no Guia do AWS IAM usuário.

Exemplo 3: Para criar um IAM usuário com um limite de permissões definido

O `create-user` comando a seguir cria um IAM usuário chamado Bob com o limite de permissões do `FullAccess AmazonS3`.

```

aws iam create-user \
  --user-name Bob \
  --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess

```

Saída:

```

{
  "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"
    }
  }
}

```

```
}  
}
```

Para obter mais informações, consulte [Limites de permissões para IAM entidades](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

create-virtual-mfa-device

O código de exemplo a seguir mostra como usar `create-virtual-mfa-device`.

AWS CLI

Para criar um MFA dispositivo virtual

Este exemplo cria um novo MFA dispositivo virtual chamado `BobsMFADevice`. Ele cria um arquivo contendo informações de bootstrap denominadas `QRCode.png` e as coloca no diretório `C:/`. O método de bootstrap usado neste exemplo é `QRCodePNG`.

```
aws iam create-virtual-mfa-device \  
  --virtual-mfa-device-name BobsMFADevice \  
  --outfile C:/QRCode.png \  
  --bootstrap-method QRCodePNG
```

Saída:

```
{  
  "VirtualMFADevice": {  
    "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"  
  }  
}
```

Para obter mais informações, consulte [Usando a autenticação multifator \(MFA\) AWS no](#) Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreateVirtualMfaDevice](#) na Referência de AWS CLI Comandos.

deactivate-mfa-device

O código de exemplo a seguir mostra como usar `deactivate-mfa-device`.

AWS CLI

Para desativar um dispositivo MFA

Esse comando desativa o MFA dispositivo virtual associado ao usuário. ARN

```
arn:aws:iam::210987654321:mfa/BobsMFADevice Bob
```

```
aws iam deactivate-mfa-device \
  --user-name Bob \
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando a autenticação multifator \(MFA\) AWS no](#) Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DeactivateMfaDevice](#) na Referência de AWS CLI Comandos.

decode-authorization-message

O código de exemplo a seguir mostra como usar `decode-authorization-message`.

AWS CLI

Para decodificar uma mensagem de falha de autorização

O `decode-authorization-message` exemplo a seguir decodifica a mensagem retornada pelo EC2 console ao tentar iniciar uma instância sem as permissões necessárias.

```
aws sts decode-authorization-message \
  --encoded-message lxzA8VEjEvu-s0TTt3PgYCXik9Yak0qsrFJGRZR98xNcyWAxwRq14xIvd-  
npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-  
mWR9hVYdvX-0zKgVOWF9pWjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpU0IjxDjzf22PF3S0E3XvIQ-  
_PE00aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8lsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX  
Xx9AYAAIr6bhcis7C__bZh4dLAAWooHFGKgfoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

A saída é formatada como uma sequência de JSON texto de linha única que você pode analisar com qualquer JSON processador de texto.

```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements  
\":{\n\"items\":[]},\n\"failures\":{\n\"items\":[]},\n\"context\":{\n\"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:InstanceType\\",\\"values\\":{\\"items
\\":[{\\"value\\":\\"t2.micro\\"}]}]},{\\"key\\":\\"ec2:RootDeviceType\\",
\\"values\\":{\\"items\\":[{\\"value\\":\\"ebs\\"}]}]},{\\"key\\":\\"aws:Region\\",\\"values
\\":{\\"items\\":[{\\"value\\":\\"us-east-2\\"}]}]},{\\"key\\":\\"aws:Service\\",\\"values
\\":{\\"items\\":[{\\"value\\":\\"ec2\\"}]}]},{\\"key\\":\\"ec2:InstanceID\\",\\"values\\":
{\\"items\\":[{\\"value\\":\\"*\\"}]}]},{\\"key\\":\\"aws:Type\\",\\"values\\":{\\"items\\":
[{\\"value\\":\\"instance\\"}]}]},{\\"key\\":\\"ec2:Tenancy\\",\\"values\\":{\\"items\\":
[{\\"value\\":\\"default\\"}]}]},{\\"key\\":\\"ec2:Region\\",\\"values\\":{\\"items\\":[{\\"value
\\":\\"us-east-2\\"}]}]},{\\"key\\":\\"aws:ARN\\",\\"values\\":{\\"items\\":[{\\"value\\":
\\arn:aws:ec2:us-east-2:403299380220:instance/*\\"}]}]}]}]}"}
}

```

Para obter mais informações, consulte [Como posso decodificar uma mensagem de falha de autorização após receber um erro "UnauthorizedOperation" durante a execução de uma EC2 instância?](#) em AWS re:POST.

- Para API obter detalhes, consulte [DecodeAuthorizationMessage](#) na Referência de AWS CLI Comandos.

delete-access-key

O código de exemplo a seguir mostra como usar delete-access-key.

AWS CLI

Para excluir uma chave de acesso para um IAM usuário

O delete-access-key comando a seguir exclui a chave de acesso especificada (ID da chave de acesso e chave de acesso secreta) para o IAM usuário nomeado Bob.

```

aws iam delete-access-key \
  --access-key-id AKIDPMS9R04H3FEXAMPLE \
  --user-name Bob

```

Este comando não produz saída.

Para listar as chaves de acesso definidas para um IAM usuário, use o `list-access-keys` comando.

Para obter mais informações, consulte [Gerenciamento de chaves de acesso para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteAccessKey](#) na Referência de AWS CLI Comandos.

delete-account-alias

O código de exemplo a seguir mostra como usar `delete-account-alias`.

AWS CLI

Como excluir um alias da conta

O comando `delete-account-alias`, apresentado a seguir, remove o alias `mycompany` para a conta atual.

```
aws iam delete-account-alias \  
  --account-alias mycompany
```

Este comando não produz saída.

Para obter mais informações, consulte [o ID AWS da sua conta e seu alias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DeleteAccountAlias](#) na Referência de AWS CLI Comandos.

delete-account-password-policy

O código de exemplo a seguir mostra como usar `delete-account-password-policy`.

AWS CLI

Excluir a política de senha da conta atual

O comando `delete-account-password-policy` a seguir remove a política de senha da conta atual.

```
aws iam delete-account-password-policy
```

Este comando não produz saída.

Para obter mais informações, consulte [Configuração de uma política de senha de conta para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteAccountPasswordPolicy](#) na Referência de AWS CLI Comandos.

delete-group-policy

O código de exemplo a seguir mostra como usar delete-group-policy.

AWS CLI

Para excluir uma política de um IAM grupo

O comando delete-group-policy, apresentado a seguir, exclui a política denominada ExamplePolicy do grupo denominado Admins.

```
aws iam delete-group-policy \  
  --group-name Admins \  
  --policy-name ExamplePolicy
```

Este comando não produz saída.

Para visualizar as políticas anexadas a um grupo, use o comando list-group-policies.

Para obter mais informações, consulte [Gerenciamento de IAM políticas](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteGroupPolicy](#) na Referência de AWS CLI Comandos.

delete-group

O código de exemplo a seguir mostra como usar delete-group.

AWS CLI

Para excluir um IAM grupo

O `delete-group` comando a seguir exclui um IAM grupo chamado `MyTestGroup`.

```
aws iam delete-group \  
  --group-name MyTestGroup
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um grupo de IAM usuários](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DeleteGroup](#) na Referência de AWS CLI Comandos.

delete-instance-profile

O código de exemplo a seguir mostra como usar `delete-instance-profile`.

AWS CLI

Como excluir um perfil de instância

O comando `delete-instance-profile`, apresentado a seguir, exclui o perfil de instância denominado `ExampleInstanceProfile`.

```
aws iam delete-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

Este comando não produz saída.

Para obter mais informações, consulte [Como usar perfis de instância](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteInstanceProfile](#) na Referência de AWS CLI Comandos.

delete-login-profile

O código de exemplo a seguir mostra como usar `delete-login-profile`.

AWS CLI

Para excluir uma senha para um IAM usuário

O `delete-login-profile` comando a seguir exclui a senha do IAM usuário chamado `Bob`.

```
aws iam delete-login-profile \  
  --user-name Bob
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de senhas para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteLoginProfile](#) na Referência de AWS CLI Comandos.

delete-open-id-connect-provider

O código de exemplo a seguir mostra como usar `delete-open-id-connect-provider`.

AWS CLI

Para excluir um provedor de IAM identidade do OpenID Connect

Este exemplo exclui o IAM OIDC provedor que se conecta ao provedor `example.oidcprovider.com`.

```
aws iam delete-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
example.oidcprovider.com
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DeleteOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

delete-policy-version

O código de exemplo a seguir mostra como usar `delete-policy-version`.

AWS CLI

Para excluir uma versão de uma política gerenciada

Este exemplo exclui a versão identificada como v2 da política cujo ARN é `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeletePolicyVersion](#) na Referência de AWS CLI Comandos.

delete-policy

O código de exemplo a seguir mostra como usar `delete-policy`.

AWS CLI

Para excluir uma IAM política

Este exemplo exclui a política cujo ARN é `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

delete-role-permissions-boundary

O código de exemplo a seguir mostra como usar `delete-role-permissions-boundary`.

AWS CLI

Para excluir um limite de permissões de uma função IAM

O `delete-role-permissions-boundary` exemplo a seguir exclui o limite de permissões para a função especificada IAM. Para aplicar um limite de permissões a um perfil, use o comando `put-role-permissions-boundary`.

```
aws iam delete-role-permissions-boundary \  
  --role-name lambda-application-role
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteRolePermissionsBoundary](#) na Referência de AWS CLI Comandos.

delete-role-policy

O código de exemplo a seguir mostra como usar `delete-role-policy`.

AWS CLI

Para remover uma política de uma IAM função

O comando `delete-role-policy`, apresentado a seguir, remove a política denominada `ExamplePolicy` do perfil denominado `Test-Role`.

```
aws iam delete-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy
```

Este comando não produz saída.

Para obter mais informações, consulte [Modificar uma função](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DeleteRolePolicy](#) na Referência de AWS CLI Comandos.

delete-role

O código de exemplo a seguir mostra como usar `delete-role`.

AWS CLI

Para excluir uma IAM função

O comando `delete-role`, apresentado a seguir, remove o perfil denominado `Test-Role`.

```
aws iam delete-role \  
  --role-name Test-Role
```

Este comando não produz saída.

Antes de poder excluir um perfil, você deve removê-lo de qualquer perfil de instância (`remove-role-from-instance-profile`), desanexar quaisquer políticas gerenciadas (`detach-role-policy`) e excluir quaisquer políticas em linha anexadas a ele (`delete-role-policy`).

Para obter mais informações, consulte [Como criar IAM funções](#) e [usar perfis de instância](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteRole](#) na Referência de AWS CLI Comandos.

delete-saml-provider

O código de exemplo a seguir mostra como usar `delete-saml-provider`.

AWS CLI

Para excluir um SAML provedor

Este exemplo exclui o provedor IAM SAML 2.0 cujo ARN é `arn:aws:iam::123456789012:saml-provider/SAMLADFSPROVIDER`.

```
aws iam delete-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSPROVIDER
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de provedores de IAM SAML identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteSAMLProvider](#) na Referência de AWS CLI Comandos.

delete-server-certificate

O código de exemplo a seguir mostra como usar `delete-server-certificate`.

AWS CLI

Para excluir um certificado de servidor da sua AWS conta

O `delete-server-certificate` comando a seguir remove o certificado de servidor especificado da sua AWS conta.

```
aws iam delete-server-certificate \  
  --server-certificate-name myUpdatedServerCertificate
```

Este comando não produz saída.

Para listar os certificados do servidor disponíveis na sua AWS conta, use o `list-server-certificates` comando.

Para obter mais informações, consulte [Gerenciamento de certificados de servidor IAM no Guia AWS IAM do usuário](#).

- Para API obter detalhes, consulte [DeleteServerCertificate](#) na Referência de AWS CLI Comandos.

`delete-service-linked-role`

O código de exemplo a seguir mostra como usar `delete-service-linked-role`.

AWS CLI

Como excluir um perfil vinculado ao serviço

O exemplo para `delete-service-linked-role`, apresentado a seguir, exclui o perfil vinculado ao serviço especificado que não é mais necessário. A exclusão acontece de forma assíncrona. É possível verificar o status da exclusão e confirmar quando ela for concluída ao usar o comando `get-service-linked-role-deletion-status`.

```
aws iam delete-service-linked-role \  
  --role-name AWSServiceRoleForLexBots
```

Saída:

```
{
```

```
"DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/  
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"  
}
```

Para obter mais informações, consulte [Usando funções vinculadas a serviços](#) no Guia do AWS IAMusuário.

- Para API obter detalhes, consulte [DeleteServiceLinkedRole](#) na Referência de AWS CLI Comandos.

delete-service-specific-credential

O código de exemplo a seguir mostra como usar `delete-service-specific-credential`.

AWS CLI

Exemplo 1: Excluir uma credencial específica do serviço para o usuário solicitante

O `delete-service-specific-credential` exemplo a seguir exclui a credencial específica do serviço especificado para o usuário que fez a solicitação. O `service-specific-credential-id` é fornecido quando você cria a credencial e você pode recuperá-la usando o `list-service-specific-credentials` comando.

```
aws iam delete-service-specific-credential \  
--service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Este comando não produz saída.

Exemplo 2: Excluir uma credencial específica do serviço para um usuário especificado

O `delete-service-specific-credential` exemplo a seguir exclui a credencial específica do serviço especificado para o usuário especificado. O `service-specific-credential-id` é fornecido quando você cria a credencial e você pode recuperá-la usando o `list-service-specific-credentials` comando.

```
aws iam delete-service-specific-credential \  
--user-name sofia \  
--service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar credenciais do Git para HTTPS conexões CodeCommit](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [DeleteServiceSpecificCredential](#) na Referência de AWS CLI Comandos.

delete-signing-certificate

O código de exemplo a seguir mostra como usar `delete-signing-certificate`.

AWS CLI

Para excluir um certificado de assinatura para um IAM usuário

O `delete-signing-certificate` comando a seguir exclui o certificado de assinatura especificado para o IAM usuário nomeado `Bob`.

```
aws iam delete-signing-certificate \  
  --user-name Bob \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE
```

Este comando não produz saída.

Para obter o ID de um certificado de assinatura, use o comando `list-signing-certificates`.

Para obter mais informações, consulte [Gerenciar certificados de assinatura](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteSigningCertificate](#) na Referência de AWS CLI Comandos.

delete-ssh-public-key

O código de exemplo a seguir mostra como usar `delete-ssh-public-key`.

AWS CLI

Para excluir uma chave SSH pública anexada a um IAM usuário

O `delete-ssh-public-key` comando a seguir exclui a chave SSH pública especificada anexada ao IAM usuário `sofia`.


```
aws iam delete-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Usar SSH teclas e SSH com CodeCommit](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteSshPublicKey](#) na Referência de AWS CLI Comandos.

delete-user-permissions-boundary

O código de exemplo a seguir mostra como usar `delete-user-permissions-boundary`.

AWS CLI

Para excluir um limite de permissões de um IAM usuário

O `delete-user-permissions-boundary` exemplo a seguir exclui o limite de permissões anexado ao IAM usuário nomeado. `intern` Para aplicar um limite de permissões a um usuário, use o comando `put-user-permissions-boundary`.

```
aws iam delete-user-permissions-boundary \  
  --user-name intern
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteUserPermissionsBoundary](#) na Referência de AWS CLI Comandos.

delete-user-policy

O código de exemplo a seguir mostra como usar `delete-user-policy`.

AWS CLI

Para remover uma política de um IAM usuário

O `delete-user-policy` comando a seguir remove a política especificada do IAM usuário nomeado Bob.

```
aws iam delete-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy
```

Este comando não produz saída.

Para obter uma lista de políticas para um IAM usuário, use o `list-user-policies` comando.

Para obter mais informações, consulte [Criação de um IAM usuário em sua AWS conta](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteUserPolicy](#) na Referência de AWS CLI Comandos.

delete-user

O código de exemplo a seguir mostra como usar `delete-user`.

AWS CLI

Para excluir um IAM usuário

O `delete-user` comando a seguir remove o IAM usuário nomeado Bob da conta atual.

```
aws iam delete-user \  
  --user-name Bob
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um IAM usuário](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

delete-virtual-mfa-device

O código de exemplo a seguir mostra como usar `delete-virtual-mfa-device`.

AWS CLI

Para remover um MFA dispositivo virtual

O `delete-virtual-mfa-device` comando a seguir remove o MFA dispositivo especificado da conta atual.

```
aws iam delete-virtual-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

Este comando não produz saída.

Para obter mais informações, consulte [Desativação de MFA dispositivos](#) no Guia do AWS IAMusuário.

- Para API obter detalhes, consulte [DeleteVirtualMfaDevice](#) na Referência de AWS CLI Comandos.

detach-group-policy

O código de exemplo a seguir mostra como usar `detach-group-policy`.

AWS CLI

Para desanexar uma política de um grupo

Este exemplo remove a política gerenciada com o ARN `arn:aws:iam::123456789012:policy/TesterAccessPolicy` do grupo chamado `Testers`.

```
aws iam detach-group-policy \  
  --group-name Testers \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de grupos de IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DetachGroupPolicy](#) na Referência de AWS CLI Comandos.

detach-role-policy

O código de exemplo a seguir mostra como usar `detach-role-policy`.

AWS CLI

Como desanexar uma política de um perfil

Este exemplo remove a política gerenciada com a ARN `arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy` da função chamada `FedTesterRole`.

```
aws iam detach-role-policy \  
  --role-name FedTesterRole \  
  --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Este comando não produz saída.

Para obter mais informações, consulte [Modificar uma função](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [DetachRolePolicy](#) na Referência de AWS CLI Comandos.

detach-user-policy

O código de exemplo a seguir mostra como usar `detach-user-policy`.

AWS CLI

Como desanexar uma política de um usuário

Este exemplo remove a política gerenciada com o ARN `arn:aws:iam::123456789012:policy/TesterPolicy` do usuário `Bob`.

```
aws iam detach-user-policy \  
  --user-name Bob \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

Este comando não produz saída.

Para obter mais informações, consulte [Alterar as permissões de um IAM usuário](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DetachUserPolicy](#) na Referência de AWS CLI Comandos.

enable-mfa-device

O código de exemplo a seguir mostra como usar `enable-mfa-device`.

AWS CLI

Para habilitar um MFA dispositivo

Depois de usar o `create-virtual-mfa-device` comando para criar um novo MFA dispositivo virtual, você pode atribuir o MFA dispositivo a um usuário. O `enable-mfa-device` exemplo a seguir atribui o MFA dispositivo com o número de série `arn:aws:iam::210987654321:mfa/BobsMFADevice` ao usuário `Bob`. O comando também sincroniza o dispositivo com AWS a inclusão dos dois primeiros códigos em sequência do MFA dispositivo virtual.

```
aws iam enable-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 789012
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando um dispositivo virtual de autenticação multifator \(MFA\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [EnableMfaDevice](#) na Referência de AWS CLI Comandos.

`generate-credential-report`

O código de exemplo a seguir mostra como usar `generate-credential-report`.

AWS CLI

Como gerar um relatório de credenciais

O exemplo a seguir tenta gerar um relatório de credenciais para a AWS conta.

```
aws iam generate-credential-report
```

Saída:

```
{  
  "State": "STARTED",  
  "Description": "No report exists. Starting a new report generation task"
```

```
}
```

Para obter mais informações, consulte [Como obter relatórios de credenciais para sua AWS conta](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GenerateCredentialReport](#) na Referência de AWS CLI Comandos.

generate-organizations-access-report

O código de exemplo a seguir mostra como usar `generate-organizations-access-report`.

AWS CLI

Exemplo 1: Para gerar um relatório de acesso para uma raiz em uma organização

O `generate-organizations-access-report` exemplo a seguir inicia um trabalho em segundo plano para criar um relatório de acesso para a raiz especificada em uma organização. Você pode exibir o relatório depois de criado executando o `get-organizations-access-report` comando.

```
aws iam generate-organizations-access-report \  
  --entity-path o-4fxmpl1t198/r-c3xb
```

Saída:

```
{  
  "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"  
}
```

Exemplo 2: Para gerar um relatório de acesso para uma conta em uma organização

O `generate-organizations-access-report` exemplo a seguir inicia um trabalho em segundo plano para criar um relatório de acesso para o ID da conta 123456789012 na organização `o-4fxmpl1t198`. Você pode exibir o relatório depois de criado executando o `get-organizations-access-report` comando.

```
aws iam generate-organizations-access-report \  
  --entity-path o-4fxmpl1t198/r-c3xb/123456789012
```

Saída:

```
{
  "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

Exemplo 3: Para gerar um relatório de acesso para uma conta em uma unidade organizacional em uma organização

O `generate-organizations-access-report` exemplo a seguir inicia um trabalho em segundo plano para criar um relatório de acesso para a ID da conta `234567890123` ou `c3xb-lmu7j2yg` na unidade organizacional da organização `o-4fxmpl1t198`. Você pode exibir o relatório depois de criado executando o `get-organizations-access-report` comando.

```
aws iam generate-organizations-access-report \
  --entity-path o-4fxmpl1t198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

Saída:

```
{
  "JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

Para obter detalhes sobre raízes e unidades organizacionais em sua organização, use os `organizations list-organizational-units-for-parent` comandos `organizations list-roots` e.

Para obter mais informações, consulte [Refinando as permissões no AWS uso das informações do último acesso](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GenerateOrganizationsAccessReport](#) na Referência de AWS CLI Comandos.

generate-service-last-accessed-details

O código de exemplo a seguir mostra como usar `generate-service-last-accessed-details`.

AWS CLI

Exemplo 1: para gerar um relatório de acesso ao serviço de uma política personalizada

O `generate-service-last-accessed-details` exemplo a seguir inicia um trabalho em segundo plano para gerar um relatório que lista os serviços acessados por IAM usuários e outras entidades com uma política personalizada chamada `intern-boundary`. Você pode exibir o relatório após a criação executando o comando `get-service-last-accessed-details`.

```
aws iam generate-service-last-accessed-details \  
--arn arn:aws:iam::123456789012:policy/intern-boundary
```

Saída:

```
{  
  "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"  
}
```

Exemplo 2: Para gerar um relatório de acesso ao serviço para a `AdministratorAccess` política AWS gerenciada

O `generate-service-last-accessed-details` exemplo a seguir inicia um trabalho em segundo plano para gerar um relatório que lista os serviços acessados por IAM usuários e outras entidades com a `AdministratorAccess` política AWS gerenciada. Você pode exibir o relatório após a criação executando o comando `get-service-last-accessed-details`.

```
aws iam generate-service-last-accessed-details \  
--arn arn:aws:iam::aws:policy/AdministratorAccess
```

Saída:

```
{  
  "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"  
}
```

Para obter mais informações, consulte [Refinando as permissões no AWS uso das informações do último acesso](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GenerateServiceLastAccessedDetails](#) na Referência de AWS CLI Comandos.

get-access-key-last-used

O código de exemplo a seguir mostra como usar `get-access-key-last-used`.

AWS CLI

Como recuperar informações sobre quando a chave de acesso especificada foi usada pela última vez

O exemplo apresentado a seguir recupera informações sobre quando a chave de acesso ABCDEXAMPLE foi usada pela última vez.

```
aws iam get-access-key-last-used \
  --access-key-id ABCDEXAMPLE
```

Saída:

```
{
  "UserName": "Bob",
  "AccessKeyLastUsed": {
    "Region": "us-east-1",
    "ServiceName": "iam",
    "LastUsedDate": "2015-06-16T22:45:00Z"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de chaves de acesso para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetAccessKeyLastUsed](#) na Referência de AWS CLI Comandos.

get-account-authorization-details

O código de exemplo a seguir mostra como usar get-account-authorization-details.

AWS CLI

Para listar AWS contas, IAM usuários, grupos, funções e políticas

O get-account-authorization-details comando a seguir retorna informações sobre todos os IAM usuários, grupos, funções e políticas na AWS conta.

```
aws iam get-account-authorization-details
```

Saída:

```
{
  "RoleDetailList": [
    {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "RoleId": "AROA1234567890EXAMPLE",
      "CreateDate": "2014-07-30T17:09:20Z",
      "InstanceProfileList": [
        {
          "InstanceProfileId": "AIPA1234567890EXAMPLE",
          "Roles": [
            {
              "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                      "Service": "ec2.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                  }
                ]
              },
              "RoleId": "AROA1234567890EXAMPLE",
              "CreateDate": "2014-07-30T17:09:20Z",
              "RoleName": "EC2role",
              "Path": "/",
              "Arn": "arn:aws:iam::123456789012:role/EC2role"
            }
          ]
        }
      ]
    }
  ]
}
```

```
    ],
    "CreateDate": "2014-07-30T17:09:20Z",
    "InstanceProfileName": "EC2role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:instance-profile/EC2role"
  }
],
"RoleName": "EC2role",
"Path": "/",
"AttachedManagedPolicies": [
  {
    "PolicyName": "AmazonS3FullAccess",
    "PolicyArn": "arn:aws:iam::aws:policy/AmazonS3FullAccess"
  },
  {
    "PolicyName": "AmazonDynamoDBFullAccess",
    "PolicyArn": "arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess"
  }
],
"RoleLastUsed": {
  "Region": "us-west-2",
  "LastUsedDate": "2019-11-13T17:30:00Z"
},
"RolePolicyList": [],
"Arn": "arn:aws:iam::123456789012:role/EC2role"
}
],
"GroupDetailList": [
  {
    "GroupId": "AIDA1234567890EXAMPLE",
    "AttachedManagedPolicies": {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    "GroupName": "Admins",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "CreateDate": "2013-10-14T18:32:24Z",
    "GroupPolicyList": []
  },
  {
    "GroupId": "AIDA1234567890EXAMPLE",
    "AttachedManagedPolicies": {
      "PolicyName": "PowerUserAccess",
```

```

        "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
    },
    "GroupName": "Dev",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:group/Dev",
    "CreateDate": "2013-10-14T18:33:55Z",
    "GroupPolicyList": []
},
{
    "GroupId": "AIDA1234567890EXAMPLE",
    "AttachedManagedPolicies": [],
    "GroupName": "Finance",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:group/Finance",
    "CreateDate": "2013-10-14T18:57:48Z",
    "GroupPolicyList": [
        {
            "PolicyName": "policygen-201310141157",
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Action": "aws-portal:*",
                        "Sid": "Stmt1381777017000",
                        "Resource": "*",
                        "Effect": "Allow"
                    }
                ]
            }
        }
    ]
}
],
"UserDetailList": [
    {
        "UserName": "Alice",
        "GroupList": [
            "Admins"
        ],
        "CreateDate": "2013-10-14T18:32:24Z",
        "UserId": "AIDA1234567890EXAMPLE",
        "UserPolicyList": [],
        "Path": "/",
        "AttachedManagedPolicies": [],

```

```
    "Arn": "arn:aws:iam::123456789012:user/Alice"
  },
  {
    "UserName": "Bob",
    "GroupList": [
      "Admins"
    ],
    "CreateDate": "2013-10-14T18:32:25Z",
    "UserId": "AIDA1234567890EXAMPLE",
    "UserPolicyList": [
      {
        "PolicyName": "DenyBillingAndIAMPolicy",
        "PolicyDocument": {
          "Version": "2012-10-17",
          "Statement": {
            "Effect": "Deny",
            "Action": [
              "aws-portal:*",
              "iam:*"
            ],
            "Resource": "*"
          }
        }
      }
    ],
    "Path": "/",
    "AttachedManagedPolicies": [],
    "Arn": "arn:aws:iam::123456789012:user/Bob"
  },
  {
    "UserName": "Charlie",
    "GroupList": [
      "Dev"
    ],
    "CreateDate": "2013-10-14T18:33:56Z",
    "UserId": "AIDA1234567890EXAMPLE",
    "UserPolicyList": [],
    "Path": "/",
    "AttachedManagedPolicies": [],
    "Arn": "arn:aws:iam::123456789012:user/Charlie"
  }
],
"Policies": [
  {
```

```
"PolicyName": "create-update-delete-set-managed-policies",
"CreateDate": "2015-02-06T19:58:34Z",
"AttachmentCount": 1,
"IsAttachable": true,
"PolicyId": "ANPA1234567890EXAMPLE",
"DefaultVersionId": "v1",
"PolicyVersionList": [
  {
    "CreateDate": "2015-02-06T19:58:34Z",
    "VersionId": "v1",
    "Document": {
      "Version": "2012-10-17",
      "Statement": {
        "Effect": "Allow",
        "Action": [
          "iam:CreatePolicy",
          "iam:CreatePolicyVersion",
          "iam>DeletePolicy",
          "iam>DeletePolicyVersion",
          "iam:GetPolicy",
          "iam:GetPolicyVersion",
          "iam:ListPolicies",
          "iam:ListPolicyVersions",
          "iam:SetDefaultPolicyVersion"
        ],
        "Resource": "*"
      }
    },
    "IsDefaultVersion": true
  }
],
"Path": "/",
"Arn": "arn:aws:iam::123456789012:policy/create-update-delete-set-
managed-policies",
"UpdateDate": "2015-02-06T19:58:34Z"
},
{
  "PolicyName": "S3-read-only-specific-bucket",
  "CreateDate": "2015-01-21T21:39:41Z",
  "AttachmentCount": 1,
  "IsAttachable": true,
  "PolicyId": "ANPA1234567890EXAMPLE",
  "DefaultVersionId": "v1",
  "PolicyVersionList": [
```

```
    {
      "CreateDate": "2015-01-21T21:39:41Z",
      "VersionId": "v1",
      "Document": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "s3:Get*",
              "s3:List*"
            ],
            "Resource": [
              "arn:aws:s3:::example-bucket",
              "arn:aws:s3:::example-bucket/*"
            ]
          }
        ]
      },
      "IsDefaultVersion": true
    }
  ],
  "Path": "/",
  "Arn": "arn:aws:iam::123456789012:policy/S3-read-only-specific-bucket",
  "UpdateDate": "2015-01-21T23:39:41Z"
},
{
  "PolicyName": "AmazonEC2FullAccess",
  "CreateDate": "2015-02-06T18:40:15Z",
  "AttachmentCount": 1,
  "IsAttachable": true,
  "PolicyId": "ANPA1234567890EXAMPLE",
  "DefaultVersionId": "v1",
  "PolicyVersionList": [
    {
      "CreateDate": "2014-10-30T20:59:46Z",
      "VersionId": "v1",
      "Document": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Action": "ec2:*",
            "Effect": "Allow",
            "Resource": "*"
          }
        ]
      }
    }
  ]
}
```

```

        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:*",
            "Resource": "*"
        }
    ]
},
    "IsDefaultVersion": true
}
],
    "Path": "/",
    "Arn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess",
    "UpdateDate": "2015-02-06T18:40:15Z"
}
],
    "Marker": "EXAMPLEkakov9BCuUNFDtxWSyetzYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
    eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
}

```

Para obter mais informações, consulte [as diretrizes AWS de auditoria de segurança](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetAccountAuthorizationDetails](#) na Referência de AWS CLI Comandos.

get-account-password-policy

O código de exemplo a seguir mostra como usar `get-account-password-policy`.

AWS CLI

Como visualizar a política de senha da conta atual

O comando `get-account-password-policy`, apresentado a seguir, exibe detalhes sobre a política de senha para a conta atual.

```
aws iam get-account-password-policy
```

Saída:

```
{
  "PasswordPolicy": {
    "AllowUsersToChangePassword": false,
    "RequireLowercaseCharacters": false,
    "RequireUppercaseCharacters": false,
    "MinimumPasswordLength": 8,
    "RequireNumbers": true,
    "RequireSymbols": true
  }
}
```

Se nenhuma política de senha estiver definida para a conta, o comando retornará um erro `NoSuchEntity`.

Para obter mais informações, consulte [Configuração de uma política de senha de conta para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetAccountPasswordPolicy](#) na Referência de AWS CLI Comandos.

`get-account-summary`

O código de exemplo a seguir mostra como usar `get-account-summary`.

AWS CLI

Para obter informações sobre o uso da IAM entidade e IAM as cotas na conta corrente

O `get-account-summary` comando a seguir retorna informações sobre o uso atual da IAM entidade e as cotas atuais da IAM entidade na conta.

```
aws iam get-account-summary
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações sobre limitações de entidades, consulte [IAM e AWS STS cotas](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetAccountSummary](#) na Referência de AWS CLI Comandos.

get-context-keys-for-custom-policy

O código de exemplo a seguir mostra como usar `get-context-keys-for-custom-policy`.

AWS CLI

Exemplo 1: Para listar as chaves de contexto referenciadas por uma ou mais JSON políticas personalizadas fornecidas como um parâmetro na linha de comando

O comando `get-context-keys-for-custom-policy` a seguir analisa cada política fornecida e lista as chaves de contexto usadas por essas políticas. Use esse comando para identificar quais valores de chave de contexto você deve fornecer para usar com êxito os comandos do simulador de políticas `simulate-custom-policy` e `simulate-policy`. Você também pode recuperar a lista de chaves de contexto usadas por todas as políticas associadas a um IAM usuário ou função usando o `get-context-keys-for-custom-policy` comando. Os valores de parâmetro que começam com `file://` instruem o comando a ler o arquivo e usar o conteúdo como o valor do parâmetro em vez do próprio nome do arquivo.

```
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"}}}'
```

Saída:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

Exemplo 2: Para listar as chaves de contexto referenciadas por uma ou mais JSON políticas personalizadas fornecidas como entrada de arquivo

O comando `get-context-keys-for-custom-policy` a seguir é igual ao exemplo anterior, exceto que as políticas são fornecidas em um arquivo e não como um parâmetro. Como o comando espera uma JSON lista de cadeias de caracteres, e não uma lista de JSON estruturas, o arquivo deve ser estruturado da seguinte forma, embora você possa reduzi-lo em uma.

```
[
  "Policy1",
```

```
"Policy2"
]
```

Assim, por exemplo, um arquivo que contém a política do exemplo anterior deve ter a aparência a seguir. Você deve escapar cada aspas duplas incorporadas dentro da string de política precedendo-as com uma barra invertida ".

```
[ {"Version": "2012-10-17", "Statement": {"Effect": "Allow", "Action": "dynamodb:*", "Resource": "arn:aws:dynamodb:us-west-2:128716708097:table/${aws:username}", "Condition": {"DateGreaterThan": {"aws:CurrentTime": "2015-08-16T12:00:00Z"}}}} ]
```

Esse arquivo pode então ser enviado ao comando a seguir.

```
aws iam get-context-keys-for-custom-policy \
  --policy-input-list file://policyfile.json
```

Saída:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

Para obter mais informações, consulte [Usando o simulador de IAM políticas \(AWS CLie AWS API\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetContextKeysForCustomPolicy](#) na Referência de AWS CLI Comandos.

get-context-keys-for-principal-policy

O código de exemplo a seguir mostra como usar `get-context-keys-for-principal-policy`.

AWS CLI

Para listar as chaves de contexto referenciadas por todas as políticas associadas a um principal IAM

O comando `get-context-keys-for-principal-policy` a seguir recupera todas as políticas anexadas à usuária `saanvi` e aos grupos dos quais ela é membro. Em seguida, ele analisa cada uma delas e lista as chaves de contexto usadas por essas políticas. Utilize esse comando para identificar quais valores de chave de contexto você deve fornecer para usar com êxito os comandos `simulate-custom-policy` e `simulate-principal-policy`. Você também pode recuperar a lista de chaves de contexto usadas por uma JSON política arbitrária usando o `get-context-keys-for-custom-policy` comando.

```
aws iam get-context-keys-for-principal-policy \  
--policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

Saída:

```
{  
  "ContextKeyNames": [  
    "aws:username",  
    "aws:CurrentTime"  
  ]  
}
```

Para obter mais informações, consulte [Usando o simulador de IAM políticas \(AWS CL e AWS API\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetContextKeysForPrincipalPolicy](#) na Referência de AWS CLI Comandos.

get-credential-report

O código de exemplo a seguir mostra como usar `get-credential-report`.

AWS CLI

Como obter um relatório de credenciais

Este exemplo abre o relatório retornado e o envia ao pipeline como uma matriz de linhas de texto.

```
aws iam get-credential-report
```

Saída:

```
{
  "GeneratedTime": "2015-06-17T19:11:50Z",
  "ReportFormat": "text/csv"
}
```

Para obter mais informações, consulte [Como obter relatórios de credenciais para sua AWS conta](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetCredentialReport](#) na Referência de AWS CLI Comandos.

get-group-policy

O código de exemplo a seguir mostra como usar `get-group-policy`.

AWS CLI

Para obter informações sobre uma política anexada a um IAM grupo

O comando `get-group-policy` a seguir obtém informações sobre a política especificada anexada ao grupo denominado `Test-Group`.

```
aws iam get-group-policy \
  --group-name Test-Group \
  --policy-name S3-ReadOnly-Policy
```

Saída:

```
{
  "GroupName": "Test-Group",
  "PolicyDocument": {
    "Statement": [
      {
        "Action": [
          "s3:Get*",
          "s3:List*"
        ],
        "Resource": "*",
        "Effect": "Allow"
      }
    ]
  },
  "PolicyName": "S3-ReadOnly-Policy"
}
```

```
}
```

Para obter mais informações, consulte [Gerenciamento de IAM políticas](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetGroupPolicy](#) na Referência de AWS CLI Comandos.

get-group

O código de exemplo a seguir mostra como usar `get-group`.

AWS CLI

Para obter um IAM grupo

Este exemplo retorna detalhes sobre o IAM grupo `Admins`.

```
aws iam get-group \  
  --group-name Admins
```

Saída:

```
{  
  "Group": {  
    "Path": "/",  
    "CreateDate": "2015-06-16T19:41:48Z",  
    "GroupId": "AIDGPM9R04H3FEXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:group/Admins",  
    "GroupName": "Admins"  
  },  
  "Users": []  
}
```

Para obter mais informações, consulte [IAM Identidades \(usuários, grupos de usuários e funções\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetGroup](#) na Referência de AWS CLI Comandos.

get-instance-profile

O código de exemplo a seguir mostra como usar `get-instance-profile`.

AWS CLI

Obter informações sobre um perfil de instância

O comando `get-instance-profile` a seguir obtém informações sobre o perfil de instância denominado `ExampleInstanceProfile`.

```
aws iam get-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

Saída:

```
{  
  "InstanceProfile": {  
    "InstanceId": "AID2MAB8DPLSRHEXAMPLE",  
    "Roles": [  
      {  
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",  
        "RoleId": "AIDGPM9R04H3FEXAMPLE",  
        "CreateDate": "2013-01-09T06:33:26Z",  
        "RoleName": "Test-Role",  
        "Path": "/",  
        "Arn": "arn:aws:iam::336924118301:role/Test-Role"  
      }  
    ],  
    "CreateDate": "2013-06-12T23:52:02Z",  
    "InstanceProfileName": "ExampleInstanceProfile",  
    "Path": "/",  
    "Arn": "arn:aws:iam::336924118301:instance-profile/ExampleInstanceProfile"  
  }  
}
```

Para obter mais informações, consulte [Como usar perfis de instância](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetInstanceProfile](#) na Referência de AWS CLI Comandos.

get-login-profile

O código de exemplo a seguir mostra como usar `get-login-profile`.

AWS CLI

Para obter informações de senha de um IAM usuário

O `get-login-profile` comando a seguir obtém informações sobre a senha do IAM usuário chamado Bob.

```
aws iam get-login-profile \  
  --user-name Bob
```

Saída:

```
{  
  "LoginProfile": {  
    "UserName": "Bob",  
    "CreateDate": "2012-09-21T23:03:39Z"  
  }  
}
```

O `get-login-profile` comando pode ser usado para verificar se um IAM usuário tem uma senha. O comando retorna um erro `NoSuchEntity` se nenhuma senha for definida para o usuário.

Não é possível visualizar uma senha com esse comando. Se a senha for perdida, você pode redefini-la (`update-login-profile`) para o usuário. Como alternativa, você pode excluir o perfil de login (`delete-login-profile`) do usuário e criar um novo (`create-login-profile`).

Para obter mais informações, consulte [Gerenciamento de senhas para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetLoginProfile](#) na Referência de AWS CLI Comandos.

get-mfa-device

O código de exemplo a seguir mostra como usar `get-mfa-device`.

AWS CLI

Para recuperar informações sobre uma chave de FIDO segurança

O exemplo de `get-mfa-device` comando a seguir recupera informações sobre a chave FIDO de segurança especificada.

```
aws iam get-mfa-device \  
  --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-  
EXAMPLEBN5FHTECLFG7EXAMPLE
```

Saída:

```
{  
  "UserName": "alice",  
  "SerialNumber": "arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-  
EXAMPLEBN5FHTECLFG7EXAMPLE",  
  "EnableDate": "2023-09-19T01:49:18+00:00",  
  "Certifications": {  
    "FIDO": "L1"  
  }  
}
```

Para obter mais informações, consulte [Usando a autenticação multifator \(MFA\) AWS no Guia do AWS IAM usuário](#).

- Para API obter detalhes, consulte [GetMfaDevice](#) na Referência de AWS CLI Comandos.

`get-open-id-connect-provider`

O código de exemplo a seguir mostra como usar `get-open-id-connect-provider`.

AWS CLI

Retornar informações sobre o provedor OpenID Connect especificado

Este exemplo retorna detalhes sobre o provedor do OpenID Connect cujo é ARN.

`arn:aws:iam::123456789012:oidc-provider/server.example.com`

```
aws iam get-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com
```

Saída:

```
{
  "Url": "server.example.com"
  "CreateDate": "2015-06-16T19:41:48Z",
  "ThumbprintList": [
    "12345abcdefghijk67890lmnopqrst987example"
  ],
  "ClientIDList": [
    "example-application-ID"
  ]
}
```

Para obter mais informações, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

get-organizations-access-report

O código de exemplo a seguir mostra como usar `get-organizations-access-report`.

AWS CLI

Para recuperar um relatório de acesso

O `get-organizations-access-report` exemplo a seguir exibe um relatório de acesso gerado anteriormente para uma entidade AWS Organizations. Para gerar um relatório, use o comando `generate-organizations-access-report`.

```
aws iam get-organizations-access-report \
  --job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

Saída:

```
{
  "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
},
...
}
```

Para obter mais informações, consulte [Refinando as permissões no AWS uso das informações do último acesso](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetOrganizationsAccessReport](#) na Referência de AWS CLI Comandos.

get-policy-version

O código de exemplo a seguir mostra como usar `get-policy-version`.

AWS CLI

Como recuperar informações sobre a versão especificada da política gerenciada especificada

Este exemplo retorna o documento de política para a versão v2 da política cujo ARN é `arn:aws:iam::123456789012:policy/MyManagedPolicy`.

```
aws iam get-policy-version \
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
  --version-id v2
```

Saída:

```
{
  "PolicyVersion": {
    "Document": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": "iam:*",
          "Resource": "*"
        }
      ]
    }
  }
}
```

```
    },
    "VersionId": "v2",
    "IsDefaultVersion": true,
    "CreateDate": "2023-04-11T00:22:54+00:00"
  }
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetPolicyVersion](#) na Referência de AWS CLI Comandos.

get-policy

O código de exemplo a seguir mostra como usar `get-policy`.

AWS CLI

Como recuperar informações sobre a política gerenciada especificada

Este exemplo retorna detalhes sobre a política gerenciada de quem ARN é `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Saída:

```
{
  "Policy": {
    "PolicyName": "MySamplePolicy",
    "CreateDate": "2015-06-17T19:23:32Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "Z27SI6FQMGNQ2EXAMPLE1",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",
    "UpdateDate": "2015-06-17T19:23:32Z"
  }
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetPolicy](#) na Referência de AWS CLI Comandos.

get-role-policy

O código de exemplo a seguir mostra como usar `get-role-policy`.

AWS CLI

Para obter informações sobre uma política associada a uma IAM função

O comando `get-role-policy` a seguir obtém informações sobre a política especificada anexada ao perfil denominado `Test-Role`.

```
aws iam get-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy
```

Saída:

```
{  
  "RoleName": "Test-Role",  
  "PolicyDocument": {  
    "Statement": [  
      {  
        "Action": [  
          "s3:ListBucket",  
          "s3:Put*",  
          "s3:Get*",  
          "s3:*MultipartUpload*"  
        ],  
        "Resource": "*",  
        "Effect": "Allow",  
        "Sid": "1"  
      }  
    ]  
  }  
  "PolicyName": "ExamplePolicy"  
}
```

Para obter mais informações, consulte [Criação de IAM funções](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetRolePolicy](#) na Referência de AWS CLI Comandos.

get-role

O código de exemplo a seguir mostra como usar `get-role`.

AWS CLI

Para obter informações sobre uma IAM função

O comando `get-role`, apresentado a seguir, obtém informações sobre o perfil denominado `Test-Role`.

```
aws iam get-role \  
  --role-name Test-Role
```

Saída:

```
{  
  "Role": {  
    "Description": "Test Role",  
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",  
    "MaxSessionDuration": 3600,  
    "RoleId": "AROA1234567890EXAMPLE",  
    "CreateDate": "2019-11-13T16:45:56Z",  
    "RoleName": "Test-Role",  
    "Path": "/",  
    "RoleLastUsed": {  
      "Region": "us-east-1",  
      "LastUsedDate": "2019-11-13T17:14:00Z"  
    },  
    "Arn": "arn:aws:iam::123456789012:role/Test-Role"  
  }  
}
```

O comando exibe a política de confiança anexada ao perfil. Para listar as políticas de permissões anexadas a um perfil, use o comando `list-role-policies`.

Para obter mais informações, consulte [Criação de IAM funções](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetRole](#) na Referência de AWS CLI Comandos.

get-saml-provider

O código de exemplo a seguir mostra como usar `get-saml-provider`.

AWS CLI

Para recuperar o metadocumento do SAML provedor

Este exemplo recupera os detalhes sobre o provedor SAML 2.0 cujo ARN é `arn:aws:iam::123456789012:saml-provider/SAMLADFS`. A resposta inclui o documento de metadados que você obteve do provedor de identidade para criar a entidade do AWS SAML provedor, bem como as datas de criação e expiração.

```
aws iam get-saml-provider \  
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Criação de provedores de IAM SAML identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetSamlProvider](#) na Referência de AWS CLI Comandos.

get-server-certificate

O código de exemplo a seguir mostra como usar `get-server-certificate`.

AWS CLI

Para obter detalhes sobre um certificado de servidor em sua AWS conta

O `get-server-certificate` comando a seguir recupera todos os detalhes sobre o certificado de servidor especificado em sua AWS conta.

```
aws iam get-server-certificate \
  --server-certificate-name myUpdatedServerCertificate
```

Saída:

```
{
  "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-----
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAGGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAGGA1UEBhMVCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
    "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCCAfICCQD6md
7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAGGA1UEBhMVCVVMxCzAJBgNVBAGT
AlldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAw
TC01BTSBDb25zb2x1MRIwEAYDVQsQQDEwLUZXN0Q21sYWxhZAdBgkqhkiG9w0BCQ
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAGGA1UEBh
MVCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxh
ZAdBgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEB
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
```

```

k60CpiwsZ3G93vUEI03IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gjpEIbb30hjZnzcqvQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
}
}

```

Para listar os certificados do servidor disponíveis na sua AWS conta, use o `list-server-certificates` comando.

Para obter mais informações, consulte [Gerenciamento de certificados de servidor IAM no Guia AWS IAM do usuário](#).

- Para API obter detalhes, consulte [GetServerCertificate](#) na Referência de AWS CLI Comandos.

get-service-last-accessed-details-with-entities

O código de exemplo a seguir mostra como usar `get-service-last-accessed-details-with-entities`.

AWS CLI

Recuperar um relatório de acesso ao serviço com detalhes de um serviço

O `get-service-last-accessed-details-with-entities` exemplo a seguir recupera um relatório que contém detalhes sobre IAM usuários e outras entidades que acessaram o serviço especificado. Para gerar um relatório, use o comando `generate-service-last-accessed-details`. Para obter uma lista de serviços acessados com namespaces, use `get-service-last-accessed-details`.

```

aws iam get-service-last-accessed-details-with-entities \
  --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \
  --service-namespace Lambda

```

Saída:

```

{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Refinando as permissões no AWS uso das informações do último acesso](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetServiceLastAccessedDetailsWithEntities](#) na Referência de AWS CLI Comandos.

get-service-last-accessed-details

O código de exemplo a seguir mostra como usar `get-service-last-accessed-details`.

AWS CLI

Recuperar um relatório de acesso ao serviço

O `get-service-last-accessed-details` exemplo a seguir recupera um relatório gerado anteriormente que lista os serviços acessados pelas IAM entidades. Para gerar um relatório, use o comando `generate-service-last-accessed-details`.

```
aws iam get-service-last-accessed-details \  
  --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

Saída:

```
{  
  "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  
    },  
  ]  
}
```

Para obter mais informações, consulte [Refinando as permissões no AWS uso das informações do último acesso](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetServiceLastAccessedDetails](#) na Referência de AWS CLI Comandos.

get-service-linked-role-deletion-status

O código de exemplo a seguir mostra como usar `get-service-linked-role-deletion-status`.

AWS CLI

Como verificar o status de uma solicitação para excluir um perfil vinculado ao serviço

O exemplo de `get-service-linked-role-deletion-status`, apresentado a seguir, exibe o status de uma solicitação anterior para excluir um perfil vinculado ao serviço. A operação de exclusão ocorre de forma assíncrona. Ao fazer a solicitação, você obtém um valor `DeletionTaskId` fornecido como parâmetro para esse comando.

```
aws iam get-service-linked-role-deletion-status \  
  --deletion-task-id
```

```
--deletion-task-id task/aws-service-role/Lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

Saída:

```
{
  "Status": "SUCCEEDED"
}
```

Para obter mais informações, consulte [Usando funções vinculadas a serviços](#) no Guia do AWS IAMusuário.

- Para API obter detalhes, consulte [GetServiceLinkedRoleDeletionStatus](#)na Referência de AWS CLI Comandos.

get-ssh-public-key

O código de exemplo a seguir mostra como usar `get-ssh-public-key`.

AWS CLI

Exemplo 1: Para recuperar uma chave SSH pública anexada a um IAM usuário em SSH formato codificado

O `get-ssh-public-key` comando a seguir recupera a chave SSH pública especificada do IAM usuáriofia. A saída está em SSH codificação.

```
aws iam get-ssh-public-key \
  --user-name sofia \
  --ssh-public-key-id APKA123456789EXAMPLE \
  --encoding SSH
```

Saída:

```
{
  "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"
  }
}

```

Exemplo 2: Para recuperar uma chave SSH pública anexada a um IAM usuário em PEM formato codificado

O `get-ssh-public-key` comando a seguir recupera a chave SSH pública especificada do IAM usuário `sofia`. A saída está em PEM codificação.

```

aws iam get-ssh-public-key \
  --user-name sofia \
  --ssh-public-key-id APKA123456789EXAMPLE \
  --encoding PEM

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Usar SSH teclas e SSH com CodeCommit](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetSshPublicKey](#) na Referência de AWS CLI Comandos.

get-user-policy

O código de exemplo a seguir mostra como usar `get-user-policy`.

AWS CLI

Para listar detalhes da política de um IAM usuário

O `get-user-policy` comando a seguir lista os detalhes da política especificada que está anexada ao nome do IAM usuário Bob.

```
aws iam get-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy
```

Saída:

```
{  
  "UserName": "Bob",  
  "PolicyName": "ExamplePolicy",  
  "PolicyDocument": {  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Action": "*",  
        "Resource": "*",  
        "Effect": "Allow"  
      }  
    ]  
  }  
}
```

Para obter uma lista de políticas para um IAM usuário, use o `list-user-policies` comando.

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetUserPolicy](#) na Referência de AWS CLI Comandos.

get-user

O código de exemplo a seguir mostra como usar `get-user`.

AWS CLI

Para obter informações sobre um IAM usuário

O `get-user` comando a seguir obtém informações sobre o IAM usuário nomeado Paulo.

```
aws iam get-user \  
  --user-name Paulo
```

Saída:

```
{
  "User": {
    "UserName": "Paulo",
    "Path": "/",
    "CreateDate": "2019-09-21T23:03:13Z",
    "UserId": "AIDA123456789EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Paulo"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetUser](#) na Referência de AWS CLI Comandos.

list-access-keys

O código de exemplo a seguir mostra como usar `list-access-keys`.

AWS CLI

Para listar a chave de acesso IDs de um IAM usuário

O `list-access-keys` comando a seguir lista as chaves IDs de acesso do IAM usuário chamado Bob.

```
aws iam list-access-keys \
  --user-name Bob
```

Saída:

```
{
  "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"
    }
]
}
```

Você não pode listar as chaves de acesso secretas IAM dos usuários. Se as chaves de acesso secretas forem perdidas, você deverá criar novas chaves de acesso usando o comando `create-access-keys`.

Para obter mais informações, consulte [Gerenciamento de chaves de acesso para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListAccessKeys](#) na Referência de AWS CLI Comandos.

list-account-aliases

O código de exemplo a seguir mostra como usar `list-account-aliases`.

AWS CLI

Como listar os aliases da conta

O comando `list-account-aliases`, apresentado a seguir, lista os aliases para a conta atual.

```
aws iam list-account-aliases
```

Saída:

```
{
  "AccountAliases": [
    "mycompany"
  ]
}
```

Para obter mais informações, consulte [o ID AWS da sua conta e seu alias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListAccountAliases](#) na Referência de AWS CLI Comandos.

list-attached-group-policies

O código de exemplo a seguir mostra como usar `list-attached-group-policies`.

AWS CLI

Para listar todas as políticas gerenciadas anexadas ao grupo especificado

Este exemplo retorna os nomes e ARNs as políticas gerenciadas que estão anexadas ao IAM grupo nomeado `Admins` na AWS conta.

```
aws iam list-attached-group-policies \
  --group-name Admins
```

Saída:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "AdministratorAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
    },
    {
      "PolicyName": "SecurityAudit",
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
    }
  ],
  "IsTruncated": false
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListAttachedGroupPolicies](#) na Referência de AWS CLI Comandos.

list-attached-role-policies

O código de exemplo a seguir mostra como usar `list-attached-role-policies`.

AWS CLI

Como listar todas as políticas gerenciadas anexadas ao perfil especificado

Esse comando retorna os nomes e ARNs as políticas gerenciadas anexadas à IAM função nomeada `SecurityAuditRole` na AWS conta.

```
aws iam list-attached-role-policies \  
  --role-name SecurityAuditRole
```

Saída:

```
{  
  "AttachedPolicies": [  
    {  
      "PolicyName": "SecurityAudit",  
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"  
    }  
  ],  
  "IsTruncated": false  
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListAttachedRolePolicies](#) na Referência de AWS CLI Comandos.

list-attached-user-policies

O código de exemplo a seguir mostra como usar `list-attached-user-policies`.

AWS CLI

Para listar todas as políticas gerenciadas anexadas ao usuário especificado

Esse comando retorna os nomes e ARNs as políticas gerenciadas do IAM usuário nomeado `Bob` na AWS conta.

```
aws iam list-attached-user-policies \  
  --user-name Bob
```

Saída:

```
{  
  "AttachedPolicies": [  
    {
```

```
    "PolicyName": "AdministratorAccess",
    "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
  },
  {
    "PolicyName": "SecurityAudit",
    "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
  }
],
"IsTruncated": false
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListAttachedUserPolicies](#) na Referência de AWS CLI Comandos.

list-entities-for-policy

O código de exemplo a seguir mostra como usar `list-entities-for-policy`.

AWS CLI

Para listar todos os usuários, grupos e perfis aos quais a política gerenciada especificada está anexada

Este exemplo retorna uma lista de IAM grupos, funções e usuários que têm a política `arn:aws:iam::123456789012:policy/TestPolicy` anexada.

```
aws iam list-entities-for-policy \
  --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

Saída:

```
{
  "PolicyGroups": [
    {
      "GroupName": "Admins",
      "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
    }
  ],
  "PolicyUsers": [
    {
```

```
        "UserName": "Alice",
        "UserId": "AIDACKCEVSQ6C2EXAMPLE"
    }
],
"PolicyRoles": [
    {
        "RoleName": "DevRole",
        "RoleId": "AR0ADBQP57FF2AEXAMPLE"
    }
],
"IsTruncated": false
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListEntitiesForPolicy](#) na Referência de AWS CLI Comandos.

list-group-policies

O código de exemplo a seguir mostra como usar `list-group-policies`.

AWS CLI

Para listar todas as políticas em linha anexadas ao grupo especificado

O `list-group-policies` comando a seguir lista os nomes das políticas em linha que estão anexadas ao IAM grupo nomeado `Admins` na conta atual.

```
aws iam list-group-policies \
  --group-name Admins
```

Saída:

```
{
  "PolicyNames": [
    "AdminRoot",
    "ExamplePolicy"
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de IAM políticas](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListGroupPolicies](#) na Referência de AWS CLI Comandos.

list-groups-for-user

O código de exemplo a seguir mostra como usar `list-groups-for-user`.

AWS CLI

Para listar os grupos aos quais um IAM usuário pertence

O `list-groups-for-user` comando a seguir exibe os grupos aos quais o IAM usuário nomeado Bob pertence.

```
aws iam list-groups-for-user \  
  --user-name Bob
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListGroupForUser](#) na Referência de AWS CLI Comandos.

list-groups

O código de exemplo a seguir mostra como usar `list-groups`.

AWS CLI

Para listar os IAM grupos da conta atual

O `list-groups` comando a seguir lista os IAM grupos na conta atual.

```
aws iam list-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListGroupns](#) na Referência de AWS CLI Comandos.

list-instance-profile-tags

O código de exemplo a seguir mostra como usar `list-instance-profile-tags`.

AWS CLI

Para listar as tags anexadas a um perfil de instância

O `list-instance-profile-tags` comando a seguir recupera a lista de tags associadas ao perfil de instância especificado.

```
aws iam list-instance-profile-tags \  
  --instance-profile-name deployment-role
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListInstanceProfileTags](#) na Referência de AWS CLI Comandos.

list-instance-profiles-for-role

O código de exemplo a seguir mostra como usar `list-instance-profiles-for-role`.

AWS CLI

Para listar os perfis de instância de uma IAM função

O comando `list-instance-profiles-for-role` a seguir lista os perfis de instância associados ao perfil `Test-Role`.

```
aws iam list-instance-profiles-for-role \  
  --role-name Test-Role
```



```
--role-name Test-Role
```

Saída:

```
{
  "InstanceProfiles": [
    {
      "InstanceId": "AIDGPM59R04H3FEXAMPLE",
      "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"
    }
  ]
}
```

Para obter mais informações, consulte [Como usar perfis de instância](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListInstanceProfilesForRole](#) na Referência de AWS CLI Comandos.

list-instance-profiles

O código de exemplo a seguir mostra como usar `list-instance-profiles`.

AWS CLI

Listar os perfis de instância da conta

O comando `list-instance-profiles` a seguir lista os perfis de instância associados à conta atual.

aws iam list-instance-profiles

Saída:

```
{
  "InstanceProfiles": [
    {
      "Path": "/",
      "InstanceProfileName": "example-dev-role",
      "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
      "Arn": "arn:aws:iam:123456789012:instance-profile/example-dev-role",
      "CreateDate": "2023-09-21T18:17:41+00:00",
      "Roles": [
        {
          "Path": "/",
          "RoleName": "example-dev-role",
          "RoleId": "AR0AJ520TH4H7LEXAMPLE",
          "Arn": "arn:aws:iam:123456789012:role/example-dev-role",
          "CreateDate": "2023-09-21T18:17:40+00:00",
          "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
              {
                "Effect": "Allow",
                "Principal": {
                  "Service": "ec2.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
              }
            ]
          }
        }
      ]
    },
    {
      "Path": "/",
      "InstanceProfileName": "example-s3-role",
      "InstanceProfileId": "AIPAJVJVNRIQFREXAMPLE",
      "Arn": "arn:aws:iam:123456789012:instance-profile/example-s3-role",
      "CreateDate": "2023-09-21T18:18:50+00:00",
      "Roles": [
        {
          "Path": "/",
```

```
"RoleName": "example-s3-role",
"RoleId": "AROAINUBC507XLEXAMPLE",
"Arn": "arn:aws:iam::123456789012:role/example-s3-role",
"CreateDate": "2023-09-21T18:18:49+00:00",
"AssumeRolePolicyDocument": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para obter mais informações, consulte [Como usar perfis de instância](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListInstanceProfiles](#) na Referência de AWS CLI Comandos.

list-mfa-device-tags

O código de exemplo a seguir mostra como usar `list-mfa-device-tags`.

AWS CLI

Para listar as etiquetas anexadas a um MFA dispositivo

O `list-mfa-device-tags` comando a seguir recupera a lista de tags associadas ao MFA dispositivo especificado.

```
aws iam list-mfa-device-tags \
  --serial-number arn:aws:iam::123456789012:mfa/alice
```

Saída:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListMfaDeviceTags](#) na Referência de AWS CLI Comandos.

list-mfa-devices

O código de exemplo a seguir mostra como usar `list-mfa-devices`.

AWS CLI

Para listar todos os MFA dispositivos de um usuário especificado

Este exemplo retorna detalhes sobre o MFA dispositivo atribuído ao IAM usuário Bob.

```
aws iam list-mfa-devices \
  --user-name Bob
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Usando a autenticação multifator \(MFA\) AWS no Guia do AWS IAM usuário](#).

- Para API obter detalhes, consulte [ListMfaDevices](#) na Referência de AWS CLI Comandos.

list-open-id-connect-provider-tags

O código de exemplo a seguir mostra como usar `list-open-id-connect-provider-tags`.

AWS CLI

Para listar as tags anexadas a um provedor de identidade compatível com OpenID Connect (OIDC)

O `list-open-id-connect-provider-tags` comando a seguir recupera a lista de tags associadas ao provedor de OIDC identidade especificado.

```

aws iam list-open-id-connect-provider-tags \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com

```

Saída:

```

{
  "Tags": [

```

```
{
  "Key": "DeptID",
  "Value": "123456"
},
{
  "Key": "Department",
  "Value": "Accounting"
}
]
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListOpenIdConnectProviderTags](#) na Referência de AWS CLI Comandos.

list-open-id-connect-providers

O código de exemplo a seguir mostra como usar `list-open-id-connect-providers`.

AWS CLI

Para listar informações sobre os provedores do OpenID Connect na conta AWS

Este exemplo retorna uma lista ARNS de todos os provedores do OpenID Connect definidos na conta atual AWS .

```
aws iam list-open-id-connect-providers
```

Saída:

```
{
  "OpenIDConnectProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
    }
  ]
}
```

Para obter mais informações, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListOpenIdConnectProviders](#) na Referência de AWS CLI Comandos.

list-policies-granting-service-access

O código de exemplo a seguir mostra como usar `list-policies-granting-service-access`.

AWS CLI

Para listar as políticas que concedem acesso principal ao serviço especificado

O `list-policies-granting-service-access` exemplo a seguir recupera a lista de políticas que concedem ao IAM usuário `sofia` acesso ao AWS CodeCommit serviço.

```
aws iam list-policies-granting-service-access \
  --arn arn:aws:iam::123456789012:user/sofia \
  --service-namespaces codecommit
```

Saída:

```
{
  "PoliciesGrantingServiceAccess": [
    {
      "ServiceNamespace": "codecommit",
      "Policies": [
        {
          "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
          "PolicyType": "INLINE",
          "EntityType": "USER",
          "EntityName": "sofia"
        }
      ]
    }
  ],
  "IsTruncated": false
}
```

Para obter mais informações, consulte [Usando IAM com CodeCommit: credenciais, SSH chaves e chaves de AWS acesso do Git](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListPoliciesGrantingServiceAccess](#) na Referência de AWS CLI Comandos.

list-policies

O código de exemplo a seguir mostra como usar `list-policies`.

AWS CLI

Para listar as políticas gerenciadas que estão disponíveis para sua AWS conta

Este exemplo retorna uma coleção das duas primeiras políticas gerenciadas disponíveis na AWS conta atual.

```
aws iam list-policies \  
  --max-items 3
```

Saída:

```
{  
  "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": "ANPAJYRXTIB4F0VS3ZXS",
    "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=="
}

```

Para obter mais informações, consulte [Políticas e permissões IAM no Guia AWS IAM do usuário](#).

- Para API obter detalhes, consulte [ListPolicies](#) na Referência de AWS CLI Comandos.

list-policy-tags

O código de exemplo a seguir mostra como usar `list-policy-tags`.

AWS CLI

Para listar as tags anexadas a uma política gerenciada

O `list-policy-tags` comando a seguir recupera a lista de tags associadas à política gerenciada especificada.

```

aws iam list-policy-tags \
  --policy-arn arn:aws:iam::123456789012:policy/billing-access

```

Saída:

```

{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",

```

```
        "Value": "Accounting"
      }
    ]
  }
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListPolicyTags](#) na Referência de AWS CLI Comandos.

list-policy-versions

O código de exemplo a seguir mostra como usar `list-policy-versions`.

AWS CLI

Listar informações sobre as versões da política gerenciada especificada

Este exemplo retorna a lista de versões disponíveis da política cujo ARN é `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam list-policy-versions \
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Saída:

```
{
  "IsTruncated": false,
  "Versions": [
    {
      "VersionId": "v2",
      "IsDefaultVersion": true,
      "CreateDate": "2015-06-02T23:19:44Z"
    },
    {
      "VersionId": "v1",
      "IsDefaultVersion": false,
      "CreateDate": "2015-06-02T22:30:47Z"
    }
  ]
}
```

Para obter mais informações, consulte [Políticas e permissões IAM no](#) Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListPolicyVersions](#) na Referência de AWS CLI Comandos.

list-role-policies

O código de exemplo a seguir mostra como usar `list-role-policies`.

AWS CLI

Para listar as políticas anexadas a uma IAM função

O `list-role-policies` comando a seguir lista os nomes das políticas de permissões para a IAM função especificada.

```
aws iam list-role-policies \  
  --role-name Test-Role
```

Saída:

```
{  
  "PolicyNames": [  
    "ExamplePolicy"  
  ]  
}
```

Para visualizar a política de confiança anexada a um perfil, use o comando `get-role`. Para visualizar os detalhes de uma política de permissões, use o comando `get-role-policy`.

Para obter mais informações, consulte [Criação de IAM funções](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListRolePolicies](#) na Referência de AWS CLI Comandos.

list-role-tags

O código de exemplo a seguir mostra como usar `list-role-tags`.

AWS CLI

Listar as tags anexadas a um perfil

O comando `list-role-tags` a seguir recupera a lista de tags associadas ao perfil especificado.

```
aws iam list-role-tags \  
  --role-name production-role
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {  
      "Key": "DeptID",  
      "Value": "12345"  
    }  
  ],  
  "IsTruncated": false  
}
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListRoleTags](#) na Referência de AWS CLI Comandos.

list-roles

O código de exemplo a seguir mostra como usar `list-roles`.

AWS CLI

Para listar IAM funções na conta atual

O `list-roles` comando a seguir lista as IAM funções da conta atual.

```
aws iam list-roles
```

Saída:

```
{  
  "Roles": [  
    {  
      "Path": "/",  
      "RoleName": "ExampleRole",
```

```
    "RoleId": "AR0AJ520TH4H7LEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
    "CreateDate": "2017-09-12T19:23:36+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    },
    "MaxSessionDuration": 3600
  },
  {
    "Path": "/example_path/",
    "RoleName": "ExampleRoleWithPath",
    "RoleId": "AR0AI4QRP7UFT7EXAMPLE",
    "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
  }
]
```

Para obter mais informações, consulte [Criação de IAM funções](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListRoles](#) na Referência de AWS CLI Comandos.

list-saml-provider-tags

O código de exemplo a seguir mostra como usar `list-saml-provider-tags`.

AWS CLI

Para listar as tags anexadas a um SAML provedor

O `list-saml-provider-tags` comando a seguir recupera a lista de tags associadas ao SAML provedor especificado.

```
aws iam list-saml-provider-tags \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListSamlProviderTags](#) na Referência de AWS CLI Comandos.

list-saml-providers

O código de exemplo a seguir mostra como usar `list-saml-providers`.

AWS CLI

Para listar os SAML provedores na AWS conta

Este exemplo recupera a lista de provedores SAML 2.0 criados na AWS conta atual.

```
aws iam list-saml-providers
```

Saída:

```
{
  "SAMLProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
      "ValidUntil": "2015-06-05T22:45:14Z",
      "CreateDate": "2015-06-05T22:45:14Z"
    }
  ]
}
```

Para obter mais informações, consulte [Criação de provedores de IAM SAML identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListSAMLProviders](#) na Referência de AWS CLI Comandos.

list-server-certificate-tags

O código de exemplo a seguir mostra como usar `list-server-certificate-tags`.

AWS CLI

Para listar as tags anexadas a um certificado de servidor

O `list-server-certificate-tags` comando a seguir recupera a lista de tags associadas ao certificado do servidor especificado.

```
aws iam list-server-certificate-tags \
  --server-certificate-name ExampleCertificate
```

Saída:

```
{
```

```
    "Tags": [
      {
        "Key": "DeptID",
        "Value": "123456"
      },
      {
        "Key": "Department",
        "Value": "Accounting"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListServerCertificateTags](#) na Referência de AWS CLI Comandos.

list-server-certificates

O código de exemplo a seguir mostra como usar `list-server-certificates`.

AWS CLI

Para listar os certificados do servidor em sua AWS conta

O `list-server-certificates` comando a seguir lista todos os certificados de servidor armazenados e disponíveis para uso em sua AWS conta.

```
aws iam list-server-certificates
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações, consulte [Gerenciamento de certificados de servidor IAM no Guia AWS IAM do usuário](#).

- Para API obter detalhes, consulte [ListServerCertificates](#) na Referência de AWS CLI Comandos.

list-service-specific-credential

O código de exemplo a seguir mostra como usar `list-service-specific-credential`.

AWS CLI

Exemplo 1: Listar as credenciais específicas do serviço para um usuário

O `list-service-specific-credentials` exemplo a seguir exibe todas as credenciais específicas do serviço atribuídas ao usuário especificado. As senhas não estão incluídas na resposta.

```
aws iam list-service-specific-credentials \
  --user-name sofia
```

Saída:

```
{
  "ServiceSpecificCredential": {
    "CreateDate": "2019-04-18T20:45:36+00:00",
    "ServiceName": "codecommit.amazonaws.com",
    "ServiceUserName": "sofia-at-123456789012",
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
    "UserName": "sofia",
    "Status": "Active"
  }
}
```

```
}  
}
```

Exemplo 2: Listar as credenciais específicas do serviço de um usuário filtradas para um serviço especificado

O `list-service-specific-credentials` exemplo a seguir exibe as credenciais específicas do serviço atribuídas ao usuário que fez a solicitação. A lista é filtrada para incluir somente as credenciais do serviço especificado. As senhas não estão incluídas na resposta.

```
aws iam list-service-specific-credentials \  
  --service-name codecommit.amazonaws.com
```

Saída:

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-04-18T20:45:36+00:00",  
    "ServiceName": "codecommit.amazonaws.com",  
    "ServiceUserName": "sofia-at-123456789012",  
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",  
    "UserName": "sofia",  
    "Status": "Active"  
  }  
}
```

Para obter mais informações, consulte [Criar credenciais do Git para HTTPS conexões CodeCommit](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [ListServiceSpecificCredential](#) na Referência de AWS CLI Comandos.

list-service-specific-credentials

O código de exemplo a seguir mostra como usar `list-service-specific-credentials`.

AWS CLI

Para recuperar uma lista de credenciais

O `list-service-specific-credentials` exemplo a seguir lista as credenciais geradas para HTTPS acesso aos AWS CodeCommit repositórios de um usuário chamado. `developer`

```
aws iam list-service-specific-credentials \  
  --user-name developer \  
  --service-name codecommit.amazonaws.com
```

Saída:

```
{  
  "ServiceSpecificCredentials": [  
    {  
      "UserName": "developer",  
      "Status": "Inactive",  
      "ServiceUserName": "developer-at-123456789012",  
      "CreateDate": "2019-10-01T04:31:41Z",  
      "ServiceSpecificCredentialId": "ACCAQFODXMPL4YFHP7DZE",  
      "ServiceName": "codecommit.amazonaws.com"  
    },  
    {  
      "UserName": "developer",  
      "Status": "Active",  
      "ServiceUserName": "developer+1-at-123456789012",  
      "CreateDate": "2019-10-01T04:31:45Z",  
      "ServiceSpecificCredentialId": "ACCAQFOXMP6VW57M7AJP",  
      "ServiceName": "codecommit.amazonaws.com"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Criar credenciais do Git para HTTPS conexões CodeCommit](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [ListServiceSpecificCredentials](#) na Referência de AWS CLI Comandos.

list-signing-certificates

O código de exemplo a seguir mostra como usar `list-signing-certificates`.

AWS CLI

Para listar os certificados de assinatura de um IAM usuário

O `list-signing-certificates` comando a seguir lista os certificados de assinatura do IAM usuário chamado Bob.

```
aws iam list-signing-certificates \  
  --user-name Bob
```

Saída:

```
{  
  "Certificates": [  
    {  
      "UserName": "Bob",  
      "Status": "Inactive",  
      "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----  
END CERTIFICATE-----",  
      "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",  
      "UploadDate": "2013-06-06T21:40:08Z"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciar certificados de assinatura](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [ListSigningCertificates](#) na Referência de AWS CLI Comandos.

list-ssh-public-keys

O código de exemplo a seguir mostra como usar `list-ssh-public-keys`.

AWS CLI

Para listar as chaves SSH públicas anexadas a um IAM usuário

O `list-ssh-public-keys` exemplo a seguir lista as chaves SSH públicas anexadas ao IAM usuário `sofia`.

```
aws iam list-ssh-public-keys \  
  --user-name sofia
```

Saída:

```
{
  "SSHPublicKeys": [
    {
      "UserName": "sofia",
      "SSHPublicKeyId": "APKA1234567890EXAMPLE",
      "Status": "Inactive",
      "UploadDate": "2019-04-18T17:04:49+00:00"
    }
  ]
}
```

Para obter mais informações, consulte [Usar SSH teclas e SSH com CodeCommit](#) no Guia AWS IAM do usuário

- Para API obter detalhes, consulte [ListSshPublicKeys](#) na Referência de AWS CLI Comandos.

list-user-policies

O código de exemplo a seguir mostra como usar `list-user-policies`.

AWS CLI

Para listar políticas para um IAM usuário

O `list-user-policies` comando a seguir lista as políticas anexadas ao nome do IAM usuário `Bob`.

```
aws iam list-user-policies \
  --user-name Bob
```

Saída:

```
{
  "PolicyNames": [
    "ExamplePolicy",
    "TestPolicy"
  ]
}
```

Para obter mais informações, consulte [Criação de um IAM usuário em sua AWS conta](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListUserPolicies](#) na Referência de AWS CLI Comandos.

list-user-tags

O código de exemplo a seguir mostra como usar `list-user-tags`.

AWS CLI

Listar as tags anexadas a um usuário

O `list-user-tags` comando a seguir recupera a lista de tags associadas ao IAM usuário especificado.

```
aws iam list-user-tags \  
  --user-name alice
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {  
      "Key": "DeptID",  
      "Value": "12345"  
    }  
  ],  
  "IsTruncated": false  
}
```

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListUserTags](#) na Referência de AWS CLI Comandos.

list-users

O código de exemplo a seguir mostra como usar `list-users`.

AWS CLI

Para listar IAM usuários

O `list-users` comando a seguir lista os IAM usuários na conta atual.

```
aws iam list-users
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Listar IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ListUsers](#) na Referência de AWS CLI Comandos.

list-virtual-mfa-devices

O código de exemplo a seguir mostra como usar `list-virtual-mfa-devices`.

AWS CLI

Para listar MFA dispositivos virtuais

O `list-virtual-mfa-devices` comando a seguir lista os MFA dispositivos virtuais que foram configurados para a conta atual.

```
aws iam list-virtual-mfa-devices
```

Saída:

```
{
  "VirtualMFADevices": [
    {
      "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
    },
    {
      "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
    }
  ]
}
```

Para obter mais informações, consulte [Habilitando um dispositivo virtual de autenticação multifator \(MFA\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ListVirtualMfaDevices](#) na Referência de AWS CLI Comandos.

put-group-policy

O código de exemplo a seguir mostra como usar `put-group-policy`.

AWS CLI

Como adicionar uma política a um grupo

O `put-group-policy` comando a seguir adiciona uma política ao IAM grupo chamado Admins.

```
aws iam put-group-policy \
  --group-name Admins \
  --policy-document file://AdminPolicy.json \
  --policy-name AdminRoot
```

Este comando não produz saída.

A política é definida como um JSON documento no AdminPolicyarquivo.json. (O nome e a extensão do arquivo não têm significado.)

Para obter mais informações, consulte [Gerenciamento de IAM políticas](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [PutGroupPolicy](#) na Referência de AWS CLI Comandos.

put-role-permissions-boundary

O código de exemplo a seguir mostra como usar `put-role-permissions-boundary`.

AWS CLI

Exemplo 1: Para aplicar um limite de permissões com base em uma política personalizada a uma IAM função

O `put-role-permissions-boundary` exemplo a seguir aplica a política personalizada `intern-boundary` chamada limite de permissões para a IAM função especificada.

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --role-name lambda-application-role
```

Este comando não produz saída.

Exemplo 2: Para aplicar um limite de permissões com base em uma política AWS gerenciada a uma IAM função

O `put-role-permissions-boundary` exemplo a seguir aplica a `PowerUserAccess` política AWS gerenciada como limite de permissões para a IAM função especificada.

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --role-name x-account-admin
```

Este comando não produz saída.

Para obter mais informações, consulte [Modificar uma função](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [PutRolePermissionsBoundary](#) na Referência de AWS CLI Comandos.

put-role-policy

O código de exemplo a seguir mostra como usar `put-role-policy`.

AWS CLI

Para anexar uma política de permissões a uma IAM função

O comando `put-role-policy`, apresentado a seguir, adiciona uma política de permissões ao perfil denominada `Test-Role`.

```
aws iam put-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

Este comando não produz saída.

A política é definida como um JSON documento no `AdminPolicyarquivo.json`. (O nome e a extensão do arquivo não têm significado.)

Para anexar uma política de confiança a um perfil, use o comando `update-assume-role-policy`.

Para obter mais informações, consulte [Modificar uma função](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [PutRolePolicy](#) na Referência de AWS CLI Comandos.

`put-user-permissions-boundary`

O código de exemplo a seguir mostra como usar `put-user-permissions-boundary`.

AWS CLI

Exemplo 1: Para aplicar um limite de permissões com base em uma política personalizada a um IAM usuário

O `put-user-permissions-boundary` exemplo a seguir aplica uma política personalizada `intern-boundary` chamada limite de permissões para o IAM usuário especificado.

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --user-name intern
```

Este comando não produz saída.

Exemplo 2: Para aplicar um limite de permissões com base em uma política AWS gerenciada a um IAM usuário

O `put-user-permissions-boundary` exemplo a seguir aplica a política AWS gerenciada chamada limite `PowerUserAccess` de permissões para o usuário especificado. IAM

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --user-name developer
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar e remover permissões de IAM identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [PutUserPermissionsBoundary](#) na Referência de AWS CLI Comandos.

put-user-policy

O código de exemplo a seguir mostra como usar `put-user-policy`.

AWS CLI

Para anexar uma política a um IAM usuário

O `put-user-policy` comando a seguir anexa uma política ao IAM usuário chamado `Bob`.

```
aws iam put-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

Este comando não produz saída.

A política é definida como um JSON documento no `AdminPolicy` arquivo `.json`. (O nome e a extensão do arquivo não têm significado.)

Para obter mais informações, consulte [Adicionar e remover permissões de IAM identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [PutUserPolicy](#) na Referência de AWS CLI Comandos.

remove-client-id-from-open-id-connect-provider

O código de exemplo a seguir mostra como usar `remove-client-id-from-open-id-connect-provider`.

AWS CLI

Para remover o ID do cliente especificado da lista de clientes IDs registrados para o provedor IAM OpenID Connect especificado

Este exemplo remove o ID `My-TestApp-3` do cliente da lista de clientes IDs associados ao IAM OIDC provedor cujo ARN é `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [RemoveClientIdFromOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

remove-role-from-instance-profile

O código de exemplo a seguir mostra como usar `remove-role-from-instance-profile`.

AWS CLI

Para remover um cargo de um perfil de instância

O comando `remove-role-from-instance-profile` a seguir remove o perfil denominado `Test-Role` do perfil de instância denominado `ExampleInstanceProfile`.

```
aws iam remove-role-from-instance-profile \
```

```
--instance-profile-name ExampleInstanceProfile \  
--role-name Test-Role
```

Para obter mais informações, consulte [Como usar perfis de instância](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [RemoveRoleFromInstanceProfile](#) na Referência de AWS CLI Comandos.

remove-user-from-group

O código de exemplo a seguir mostra como usar `remove-user-from-group`.

AWS CLI

Para remover um usuário de um IAM grupo

O `remove-user-from-group` comando a seguir remove o usuário nomeado Bob do IAM grupo nomeado Admins.

```
aws iam remove-user-from-group \  
--user-name Bob \  
--group-name Admins
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar e remover usuários em um IAM grupo](#) de usuários no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [RemoveUserFromGroup](#) na Referência de AWS CLI Comandos.

reset-service-specific-credential

O código de exemplo a seguir mostra como usar `reset-service-specific-credential`.

AWS CLI

Exemplo 1: redefinir a senha de uma credencial específica do serviço anexada ao usuário que fez a solicitação

O `reset-service-specific-credential` exemplo a seguir gera uma nova senha criptograficamente forte para a credencial específica do serviço especificado anexada ao usuário que está fazendo a solicitação.

```
aws iam reset-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Saída:

```
{  
  "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"  
  }  
}
```

Exemplo 2: redefinir a senha de uma credencial específica do serviço anexada a um usuário especificado

O `reset-service-specific-credential` exemplo a seguir gera uma nova senha criptograficamente forte para uma credencial específica do serviço anexada ao usuário especificado.

```
aws iam reset-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Criar credenciais do Git para HTTPS conexões CodeCommit](#) no Guia do AWS CodeCommit usuário.

- Para API obter detalhes, consulte [ResetServiceSpecificCredential](#) na Referência de AWS CLI Comandos.

resync-mfa-device

O código de exemplo a seguir mostra como usar `resync-mfa-device`.

AWS CLI

Para sincronizar um dispositivo MFA

O `resync-mfa-device` exemplo a seguir sincroniza o MFA dispositivo que está associado ao IAM usuário Bob e que ARN está `arn:aws:iam::123456789012:mfa/BobsMFADevice` com um programa autenticador que forneceu os dois códigos de autenticação.

```
aws iam resync-mfa-device \
  --user-name Bob \
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
  --authentication-code1 123456 \
  --authentication-code2 987654
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando a autenticação multifator \(MFA\) AWS no](#) Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ResyncMfaDevice](#) na Referência de AWS CLI Comandos.

set-default-policy-version

O código de exemplo a seguir mostra como usar `set-default-policy-version`.

AWS CLI

Para definir a versão especificada da política especificada como a versão da política padrão.

Este exemplo define a v2 versão da política que ARN é `arn:aws:iam::123456789012:policy/MyPolicy` a versão ativa padrão.

```
aws iam set-default-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Para obter mais informações, consulte [Políticas e permissões IAM no Guia AWS IAM do usuário](#).

- Para API obter detalhes, consulte [SetDefaultPolicyVersion](#) na Referência de AWS CLI Comandos.

set-security-token-service-preferences

O código de exemplo a seguir mostra como usar `set-security-token-service-preferences`.

AWS CLI

Para definir a versão global do token de endpoint

O `set-security-token-service-preferences` exemplo a seguir configura STS a Amazon para usar tokens da versão 2 ao se autenticar no endpoint global.

```
aws iam set-security-token-service-preferences \  
  --global-endpoint-token-version v2Token
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciando AWS STS em uma AWS região](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [SetSecurityTokenServicePreferences](#) na Referência de AWS CLI Comandos.

simulate-custom-policy

O código de exemplo a seguir mostra como usar `simulate-custom-policy`.

AWS CLI

Exemplo 1: Para simular os efeitos de todas as IAM políticas associadas a um IAM usuário ou função

A seguir, `simulate-custom-policy` mostramos como fornecer a política e definir valores de variáveis e simular uma API chamada para ver se ela é permitida ou negada. O exemplo a seguir mostra uma política que permite o acesso ao banco de dados somente após uma data e hora especificadas. A simulação é bem-sucedida porque as ações simuladas e a `aws:CurrentTime` variável especificada atendem aos requisitos da política.

```
aws iam simulate-custom-policy \
  --policy-input-list '{"Version":"2012-10-17","Statement":
{"Effect":"Allow","Action":"dynamodb:*","Resource":"*","Condition":
{"DateGreaterThan":{"aws:CurrentTime":"2018-08-16T12:00:00Z"}}}' \
  --action-names dynamodb>CreateBackup \
  --context-
entries "ContextKeyName='aws:CurrentTime',ContextKeyValues='2019-04-25T11:00:00Z',ContextKey
```

Saída:

```
{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb>CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "allowed",
      "MatchedStatements": [
        {
          "SourcePolicyId": "PolicyInputList.1",
          "StartPosition": {
            "Line": 1,
            "Column": 38
          },
          "EndPosition": {
            "Line": 1,
            "Column": 167
          }
        }
      ],
      "MissingContextValues": []
    }
  ]
}
```

```
]
}
```

Exemplo 2: Para simular um comando proibido pela política

O `simulate-custom-policy` exemplo a seguir mostra os resultados da simulação de um comando proibido pela política. Neste exemplo, a data fornecida é anterior à exigida pela condição da política.

```
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
```

Saída:

```
{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb:CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "implicitDeny",
      "MatchedStatements": [],
      "MissingContextValues": []
    }
  ]
}
```

Para obter mais informações, consulte [Testando IAM políticas com o simulador de IAM políticas](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [SimulateCustomPolicy](#) na Referência de AWS CLI Comandos.

`simulate-principal-policy`

O código de exemplo a seguir mostra como usar `simulate-principal-policy`.

AWS CLI

Exemplo 1: Para simular os efeitos de uma política arbitrária IAM

Veja a `simulate-principal-policy` seguir como simular um usuário chamando uma API ação e determinando se as políticas associadas a esse usuário permitem ou negam a ação. No exemplo a seguir, o usuário tem uma política que permite somente a `codecommit:ListRepositories` ação.

```
aws iam simulate-principal-policy \  
  --policy-source-arn arn:aws:iam::123456789012:user/alejandro \  
  --action-names codecommit:ListRepositories
```

Saída:

```
{  
  "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": []  
    }  
  ]  
}
```

Exemplo 2: Para simular os efeitos de um comando proibido

O `simulate-custom-policy` exemplo a seguir mostra os resultados da simulação de um comando proibido por uma das políticas do usuário. No exemplo a seguir, o usuário tem uma política que permite o acesso a um banco de dados do DynamoDB somente após uma determinada data e hora. A simulação mostra o usuário tentando acessar o banco de dados com um `aws:CurrentTime` valor anterior ao permitido pela condição da política.

```
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
```

Saída:

```
{  
  "EvaluationResults": [  
    {  
      "EvalActionName": "dynamodb:CreateBackup",  
      "EvalResourceName": "*",  
      "EvalDecision": "implicitDeny",  
      "MatchedStatements": [],  
      "MissingContextValues": []  
    }  
  ]  
}
```

Para obter mais informações, consulte [Testando IAM políticas com o simulador de IAM políticas](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [SimulatePrincipalPolicy](#) na Referência de AWS CLI Comandos.

tag-instance-profile

O código de exemplo a seguir mostra como usar `tag-instance-profile`.

AWS CLI

Para adicionar uma tag a um perfil de instância

O `tag-instance-profile` comando a seguir adiciona uma tag com o nome do Departamento ao perfil de instância especificado.

```
aws iam tag-instance-profile \  
  --instance-profile-name deployment-role \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagInstanceProfile](#) na Referência de AWS CLI Comandos.

tag-mfa-device

O código de exemplo a seguir mostra como usar tag-mfa-device.

AWS CLI

Para adicionar uma tag a um MFA dispositivo

O tag-mfa-device comando a seguir adiciona uma tag com o nome do Departamento ao MFA dispositivo especificado.

```
aws iam tag-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagMfaDevice](#) na Referência de AWS CLI Comandos.

tag-open-id-connect-provider

O código de exemplo a seguir mostra como usar tag-open-id-connect-provider.

AWS CLI

Para adicionar uma tag a um provedor de identidade compatível com OpenID Connect (OIDC)

O tag-open-id-connect-provider comando a seguir adiciona uma tag com o nome do Departamento ao provedor de OIDC identidade especificado.

```
aws iam tag-open-id-connect-provider \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

```
--open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
--tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

tag-policy

O código de exemplo a seguir mostra como usar tag-policy.

AWS CLI

Para adicionar uma tag a uma política gerenciada pelo cliente

O tag-policy comando a seguir adiciona uma tag com o nome do Departamento à política gerenciada pelo cliente especificada.

```
aws iam tag-policy \  
--policy-arn arn:aws:iam::123456789012:policy/billing-access \  
--tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagPolicy](#) na Referência de AWS CLI Comandos.

tag-role

O código de exemplo a seguir mostra como usar tag-role.

AWS CLI

Adicionar uma tag a um perfil

O comando tag-role a seguir adiciona uma tag com o nome do Departamento ao perfil especificado.

```
aws iam tag-role --role-name my-role \  
  --tags '{"Key": "Department", "Value": "Accounting"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagRole](#) na Referência de AWS CLI Comandos.

tag-saml-provider

O código de exemplo a seguir mostra como usar tag-saml-provider.

AWS CLI

Para adicionar uma tag a um SAML provedor

O tag-saml-provider comando a seguir adiciona uma tag com o nome do Departamento ao SAML provedor especificado.

```
aws iam tag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagSamlProvider](#) na Referência de AWS CLI Comandos.

tag-server-certificate

O código de exemplo a seguir mostra como usar tag-server-certificate.

AWS CLI

Para adicionar uma tag a um certificado de servidor

O tag-saml-provider comando a seguir adiciona uma tag com o nome do Departamento ao certificado do servidor especificado.

```
aws iam tag-server-certificate \  
  --server-certificate-arn arn:aws:iam::123456789012:server-certificate/MyCertificate \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

```
--server-certificate-name ExampleCertificate \  
--tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagServerCertificate](#) na Referência de AWS CLI Comandos.

tag-user

O código de exemplo a seguir mostra como usar `tag-user`.

AWS CLI

Adicionar uma tag a um usuário

O comando `tag-user` a seguir adiciona uma tag com o Departamento associado ao usuário especificado.

```
aws iam tag-user \  
--user-name alice \  
--tags '{"Key": "Department", "Value": "Accounting"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [TagUser](#) na Referência de AWS CLI Comandos.

untag-instance-profile

O código de exemplo a seguir mostra como usar `untag-instance-profile`.

AWS CLI

Para remover uma tag de um perfil de instância

O `untag-instance-profile` comando a seguir remove qualquer tag com o nome de chave “Departamento” do perfil de instância especificado.

```
aws iam untag-instance-profile \  

```



```
--instance-profile-name deployment-role \  
--tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagInstanceProfile](#) na Referência de AWS CLI Comandos.

untag-mfa-device

O código de exemplo a seguir mostra como usar `untag-mfa-device`.

AWS CLI

Para remover uma etiqueta de um MFA dispositivo

O `untag-mfa-device` comando a seguir remove qualquer tag com o nome de chave 'Departamento' do MFA dispositivo especificado.

```
aws iam untag-mfa-device \  
--serial-number arn:aws:iam::123456789012:mfa/alice \  
--tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagMfaDevice](#) na Referência de AWS CLI Comandos.

untag-open-id-connect-provider

O código de exemplo a seguir mostra como usar `untag-open-id-connect-provider`.

AWS CLI

Para remover uma tag de um provedor de OIDC identidade

O `untag-open-id-connect-provider` comando a seguir remove qualquer tag com o nome de chave 'Departamento' do provedor de OIDC identidade especificado.

```
aws iam untag-open-id-connect-provider \  

```

```
--open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
--tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagOpenIdConnectProvider](#) na Referência de AWS CLI Comandos.

untag-policy

O código de exemplo a seguir mostra como usar `untag-policy`.

AWS CLI

Para remover uma tag de uma política gerenciada pelo cliente

O `untag-policy` comando a seguir remove qualquer tag com o nome de chave "Departamento" da política gerenciada pelo cliente especificada.

```
aws iam untag-policy \  
--policy-arn arn:aws:iam::452925170507:policy/billing-access \  
--tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagPolicy](#) na Referência de AWS CLI Comandos.

untag-role

O código de exemplo a seguir mostra como usar `untag-role`.

AWS CLI

Remover uma tag de um perfil

O comando `untag-role` a seguir remove qualquer tag com o nome de chave 'Department' do perfil especificado.

```
aws iam untag-role \  
  --role-name my-role \  
  --tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagRole](#) na Referência de AWS CLI Comandos.

untag-saml-provider

O código de exemplo a seguir mostra como usar `untag-saml-provider`.

AWS CLI

Para remover uma tag de um SAML provedor

O `untag-saml-provider` comando a seguir remove qualquer tag com o nome de chave "Departamento" do perfil de instância especificado.

```
aws iam untag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagSamlProvider](#) na Referência de AWS CLI Comandos.

untag-server-certificate

O código de exemplo a seguir mostra como usar `untag-server-certificate`.

AWS CLI

Para remover uma tag de um certificado de servidor

O `untag-server-certificate` comando a seguir remove qualquer tag com o nome de chave 'Departamento' do certificado do servidor especificado.

```
aws iam untag-server-certificate \  
  --server-certificate-name ExampleCertificate \  
  --tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagServerCertificate](#) na Referência de AWS CLI Comandos.

untag-user

O código de exemplo a seguir mostra como usar untag-user.

AWS CLI

Remover uma tag de um usuário

O comando untag-user a seguir remove qualquer tag com o nome de chave 'Department' do usuário especificado.

```
aws iam untag-user \  
  --user-name alice \  
  --tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar IAM recursos](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UntagUser](#) na Referência de AWS CLI Comandos.

update-access-key

O código de exemplo a seguir mostra como usar update-access-key.

AWS CLI

Para ativar ou desativar uma chave de acesso para um usuário IAM

O update-access-key comando a seguir desativa a chave de acesso especificada (ID da chave de acesso e chave de acesso secreta) para o IAM usuário nomeado. Bob

```
aws iam update-access-key \  
  --access-key-id AKIAIOSFODNN7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

Este comando não produz saída.

Desativar a chave significa que ela não pode ser usada para acesso programático a. AWS No entanto, a chave continua disponível e pode ser ativada novamente.

Para obter mais informações, consulte [Gerenciamento de chaves de acesso para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateAccessKey](#) na Referência de AWS CLI Comandos.

update-account-password-policy

O código de exemplo a seguir mostra como usar `update-account-password-policy`.

AWS CLI

Definir ou alterar a política de senha da conta atual

O comando `update-account-password-policy` a seguir define a política de senha para exigir um mínimo de oito caracteres e um ou mais números na senha.

```
aws iam update-account-password-policy \  
  --minimum-password-length 8 \  
  --require-numbers
```

Este comando não produz saída.

As alterações na política de senha de uma conta afetam todas as novas senhas criadas para IAM os usuários na conta. As alterações na política de senha não afetam as senhas existentes.

Para obter mais informações, consulte [Configuração de uma política de senha de conta para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateAccountPasswordPolicy](#) na Referência de AWS CLI Comandos.

update-assume-role-policy

O código de exemplo a seguir mostra como usar `update-assume-role-policy`.

AWS CLI

Para atualizar a política de confiança de uma IAM função

O comando `update-assume-role-policy` a seguir atualiza a política de confiança do perfil denominado `Test-Role`.

```
aws iam update-assume-role-policy \  
  --role-name Test-Role \  
  --policy-document file://Test-Role-Trust-Policy.json
```

Este comando não produz saída.

A política de confiança é definida como um JSON documento no arquivo `Test-role-trust-policy.json`. (O nome e a extensão do arquivo não têm significado.) A política de confiança deve especificar uma entidade principal.

Para atualizar a política de permissões de um perfil, use o comando `put-role-policy`.

Para obter mais informações, consulte [Criação de IAM funções](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateAssumeRolePolicy](#) na Referência de AWS CLI Comandos.

update-group

O código de exemplo a seguir mostra como usar `update-group`.

AWS CLI

Para renomear um grupo IAM

O `update-group` comando a seguir altera o nome do IAM grupo `Test` para `Test-1`.

```
aws iam update-group \  
  --group-name Test \  
  --new-group-name Test-1
```

Este comando não produz saída.

Para obter mais informações, consulte [Renomear um grupo de IAM usuários](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UpdateGroup](#) na Referência de AWS CLI Comandos.

update-login-profile

O código de exemplo a seguir mostra como usar `update-login-profile`.

AWS CLI

Para atualizar a senha de um IAM usuário

O `update-login-profile` comando a seguir cria uma nova senha para o IAM usuário chamado Bob.

```
aws iam update-login-profile \  
  --user-name Bob \  
  --password <password>
```

Este comando não produz saída.

Para definir uma política de senha da conta, use o comando `update-account-password-policy`. Se a nova senha violar a política de senha da conta, o comando retornará um erro `PasswordPolicyViolation`.

Se a política de senha da conta permitir, IAM os usuários poderão alterar suas próprias senhas usando o `change-password` comando.

Armazene a senha em um lugar seguro. Se a senha for perdida, não será possível recuperá-la e você deverá criar uma nova usando o comando `create-login-profile`.

Para obter mais informações, consulte [Gerenciamento de senhas para IAM usuários](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateLoginProfile](#) na Referência de AWS CLI Comandos.

update-open-id-connect-provider-thumbprint

O código de exemplo a seguir mostra como usar `update-open-id-connect-provider-thumbprint`.

AWS CLI

Para substituir a lista existente de impressões digitais do certificado de servidor por uma nova

Este exemplo atualiza a lista de impressões digitais do certificado para o OIDC provedor que `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` deve usar uma nova impressão digital. ARN

```
aws iam update-open-id-connect-provider-thumbprint \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
example.oidcprovider.com \  
  --thumbprint-list 7359755EXAMPLEEabc3060bce3EXAMPLEEec4542a3
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de provedores de identidade OpenID Connect \(OIDC\)](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UpdateOpenIdConnectProviderThumbprint](#) na Referência de AWS CLI Comandos.

update-role-description

O código de exemplo a seguir mostra como usar `update-role-description`.

AWS CLI

Para alterar a descrição de uma IAM função

O `update-role` comando a seguir altera a descrição da IAM função `production-role` para `Main production role`.

```
aws iam update-role-description \  
  --role-name production-role \  
  --description 'Main production role'
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Modificar uma função](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UpdateRoleDescription](#) na Referência de AWS CLI Comandos.

update-role

O código de exemplo a seguir mostra como usar `update-role`.

AWS CLI

Para alterar a descrição de uma IAM função ou a duração da sessão

O `update-role` comando a seguir altera a descrição da IAM função `production-role` `Main production role` e define a duração máxima da sessão para 12 horas.

```
aws iam update-role \
  --role-name production-role \
  --description 'Main production role' \
  --max-session-duration 43200
```

Este comando não produz saída.

Para obter mais informações, consulte [Modificar uma função](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UpdateRole](#) na Referência de AWS CLI Comandos.

update-saml-provider

O código de exemplo a seguir mostra como usar `update-saml-provider`.

AWS CLI

Para atualizar o documento de metadados de um provedor existente SAML

Este exemplo atualiza o SAML provedor em IAM cujo ARN está `arn:aws:iam::123456789012:saml-provider/SAMLADFS` com um novo documento de SAML metadados do arquivo `SAMLMetaData.xml`.

```
aws iam update-saml-provider \  
  --saml-metadata-document file://SAMLMetaData.xml \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Saída:

```
{  
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"  
}
```

Para obter mais informações, consulte [Criação de provedores de IAM SAML identidade](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateSamlProvider](#) na Referência de AWS CLI Comandos.

update-server-certificate

O código de exemplo a seguir mostra como usar `update-server-certificate`.

AWS CLI

Para alterar o caminho ou o nome de um certificado de servidor em sua AWS conta

O comando `update-server-certificate`, apresentado a seguir, altera o nome do certificado de `myServerCertificate` para `myUpdatedServerCertificate`. Ele também altera o

caminho para que `/cloudfront/` possa ser acessado pelo CloudFront serviço da Amazon. Este comando não produz saída. É possível visualizar os resultados da atualização ao executar o comando `list-server-certificates`.

```
aws-iam update-server-certificate \  
  --server-certificate-name myServerCertificate \  
  --new-server-certificate-name myUpdatedServerCertificate \  
  --new-path /cloudfront/
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de certificados de servidor IAM no Guia AWS IAM do usuário](#).

- Para API obter detalhes, consulte [UpdateServerCertificate](#) na Referência de AWS CLI Comandos.

update-service-specific-credential

O código de exemplo a seguir mostra como usar `update-service-specific-credential`.

AWS CLI

Exemplo 1: Para atualizar o status da credencial específica do serviço do usuário solicitante

O `update-service-specific-credential` exemplo a seguir altera o status da credencial especificada para o usuário que está fazendo a Inactive solicitação.

```
aws iam update-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

Este comando não produz saída.

Exemplo 2: Para atualizar o status da credencial específica do serviço de um usuário especificado

O `update-service-specific-credential` exemplo a seguir altera o status da credencial do usuário especificado para Inativo.

```
aws iam update-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

```
--user-name sofia \  
--service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
--status Inactive
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar credenciais do Git para HTTPS conexões CodeCommit](#) no Guia do usuário AWS CodeCommit

- Para API obter detalhes, consulte [UpdateServiceSpecificCredential](#) na Referência de AWS CLI Comandos.

update-signing-certificate

O código de exemplo a seguir mostra como usar `update-signing-certificate`.

AWS CLI

Para ativar ou desativar um certificado de assinatura para um usuário IAM

O `update-signing-certificate` comando a seguir desativa o certificado de assinatura especificado para o IAM usuário nomeado. Bob

```
aws iam update-signing-certificate \  
--certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE \  
--status Inactive \  
--user-name Bob
```

Para obter o ID de um certificado de assinatura, use o comando `list-signing-certificates`.

Para obter mais informações, consulte [Gerenciar certificados de assinatura](#) no Guia EC2 do usuário da Amazon.

- Para API obter detalhes, consulte [UpdateSigningCertificate](#) na Referência de AWS CLI Comandos.

update-ssh-public-key

O código de exemplo a seguir mostra como usar `update-ssh-public-key`.

AWS CLI

Para alterar o status de uma chave SSH pública

O `update-ssh-public-key` comando a seguir altera o status da chave pública especificada para `Inactive`.

```
aws iam update-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA1234567890EXAMPLE \  
  --status Inactive
```

Este comando não produz saída.

Para obter mais informações, consulte [Usar SSH teclas e SSH com CodeCommit](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateSshPublicKey](#) na Referência de AWS CLI Comandos.

update-user

O código de exemplo a seguir mostra como usar `update-user`.

AWS CLI

Para alterar o nome de um IAM usuário

O `update-user` comando a seguir altera o nome do IAM usuário Bob para Robert.

```
aws iam update-user \  
  --user-name Bob \  
  --new-user-name Robert
```

Este comando não produz saída.

Para obter mais informações, consulte [Renomear um grupo de IAM usuários](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [UpdateUser](#) na Referência de AWS CLI Comandos.

upload-server-certificate

O código de exemplo a seguir mostra como usar `upload-server-certificate`.

AWS CLI

Para fazer upload de um certificado de servidor para sua AWS conta

O `upload-server-certificate` comando a seguir carrega um certificado de servidor em sua AWS conta. Neste exemplo, o certificado está no arquivo `public_key_cert_file.pem`, a chave privada associada está no arquivo `my_private_key.pem` e a cadeia de certificados fornecida pela autoridade de certificação (CA) está no arquivo `my_certificate_chain_file.pem`. Quando o upload do arquivo for concluído, ele estará disponível com o nome `myServerCertificate`. Os parâmetros que começam com `file://` informam ao comando para ler o conteúdo do arquivo e usá-lo como valor do parâmetro em vez do próprio nome do arquivo.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Criação, upload e exclusão de certificados de servidor no guia de uso IAM.

- Para API obter detalhes, consulte [UploadServerCertificate](#) na Referência de AWS CLI Comandos.

upload-signing-certificate

O código de exemplo a seguir mostra como usar `upload-signing-certificate`.

AWS CLI

Para carregar um certificado de assinatura para um IAM usuário

O `upload-signing-certificate` comando a seguir carrega um certificado de assinatura para o IAM usuário chamado Bob.

```
aws iam upload-signing-certificate \  
  --user-name Bob \  
  --certificate-body file://certificate.pem
```

Saída:

```
{  
  "Certificate": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END  
CERTIFICATE-----",  
    "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",  
    "UploadDate": "2013-06-06T21:40:08.121Z"  
  }  
}
```

O certificado está em um arquivo chamado `certificate.pem` no formato PEM

Para obter mais informações, consulte Criação e upload de um certificado de assinatura de usuário no IAM guia de uso.

- Para API obter detalhes, consulte [UploadSigningCertificate](#) na Referência de AWS CLI Comandos.

upload-ssh-public-key

O código de exemplo a seguir mostra como usar `upload-ssh-public-key`.

AWS CLI

Para carregar uma chave SSH pública e associá-la a um usuário

O `upload-ssh-public-key` comando a seguir carrega a chave pública encontrada no arquivo `sshkey.pub` e a anexa ao usuário. `sofia`

```
aws iam upload-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-body file://sshkey.pub
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações sobre como gerar chaves em um formato adequado para esse comando, consulte [SSHLinux, macOS ou Unix: Configurar as chaves públicas e privadas para Git CodeCommit](#) e/ou [Windows: Configurar as chaves públicas SSH e privadas para Git e no Guia do Usuário. CodeCommit AWS CodeCommit](#)

- Para API obter detalhes, consulte [UploadSshPublicKey](#) na Referência de AWS CLI Comandos.

IAM Exemplos do Access Analyzer usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o IAM Access Analyzer.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

apply-archive-rule

O código de exemplo a seguir mostra como usar `apply-archive-rule`.

AWS CLI

Para aplicar uma regra de arquivamento às descobertas existentes que atendam aos critérios da regra de arquivamento

O `apply-archive-rule` exemplo a seguir aplica uma regra de arquivamento às descobertas existentes que atendem aos critérios da regra de arquivamento.

```
aws accessanalyzer apply-archive-rule \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule
```

Este comando não produz saída.

Para obter mais informações, consulte [Regras de arquivamento](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [ApplyArchiveRule](#) na Referência de AWS CLI Comandos.

cancel-policy-generation

O código de exemplo a seguir mostra como usar `cancel-policy-generation`.

AWS CLI

Para cancelar a geração da política solicitada

O `cancel-policy-generation` exemplo a seguir cancela a ID do job de geração de política solicitada.

```
aws accessanalyzer cancel-policy-generation \  
  --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

Este comando não produz saída.

Para obter mais informações, consulte [Geração de políticas do IAM Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [CancelPolicyGeneration](#) na Referência de AWS CLI Comandos.

check-access-not-granted

O código de exemplo a seguir mostra como usar `check-access-not-granted`.

AWS CLI

Para verificar se o acesso especificado não é permitido por uma política

O `check-access-not-granted` exemplo a seguir verifica se o acesso especificado não é permitido por uma política.

```
aws accessanalyzer check-access-not-granted \  
  --policy-document file://myfile.json \  
  --access actions="s3:DeleteBucket","s3:GetBucketLocation" \  
  --policy-type IDENTITY_POLICY
```

Conteúdo de `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/*"  
      ]  
    }  
  ]  
}
```

Saída:

```
{
```

```
"result": "PASS",
"message": "The policy document does not grant access to perform the listed
actions."
}
```

Para obter mais informações, consulte [Visualizando o acesso com o IAM Access Analyzer APIs](#) no Guia do AWS IAMUsuário.

- Para API obter detalhes, consulte [CheckAccessNotGranted](#) na Referência de AWS CLI Comandos.

check-no-new-access

O código de exemplo a seguir mostra como usar check-no-new-access.

AWS CLI

Para verificar se um novo acesso é permitido para uma política atualizada em comparação com a política existente

O check-no-new-access exemplo a seguir verifica se um novo acesso é permitido para uma política atualizada em comparação com a política existente.

```
aws accessanalyzer check-no-new-access \
  --existing-policy-document file://existing-policy.json \
  --new-policy-document file://new-policy.json \
  --policy-type IDENTITY_POLICY
```

Conteúdo de 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/*"
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Conteúdo de 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/*"
      ]
    }
  ]
}
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Visualizando o acesso com o IAM Access Analyzer APIs](#) no Guia do AWS IAMUsuário.

- Para API obter detalhes, consulte [CheckNoNewAccess](#) na Referência de AWS CLI Comandos.

create-access-preview

O código de exemplo a seguir mostra como usar `create-access-preview`.

AWS CLI

Para criar uma visualização prévia de acesso que permita que você visualize as descobertas do IAM Access Analyzer para seu recurso antes de implantar as permissões do recurso

O `create-access-preview` exemplo a seguir cria uma visualização prévia de acesso que permite que você visualize as descobertas do IAM Access Analyzer para seu recurso antes de implantar permissões de recursos em sua AWS conta.

```
aws accessanalyzer create-access-preview \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --configurations file://myfile.json
```

Conteúdo de `myfile.json`:

```
{  
  "arn:aws:s3:::DOC-EXAMPLE-BUCKET": {  
    "s3Bucket": {  
      "bucketPolicy": "{\n\"Version\":\n\"2012-10-17\",\n\"Statement\":[\n{\n\"Effect\n\":\n\"Allow\",\n\"Principal\":{\n\"AWS\":[\n\"arn:aws:iam::111122223333:root\n\"}],\n\"Action\n\":[\n\"s3:PutObject\",\n\"s3:PutObjectAcl\n\"],\n\"Resource\n\":\n\"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*\n\"}}]",  
      "bucketPublicAccessBlock": {  
        "ignorePublicAcls": true,  
        "restrictPublicBuckets": true  
      },  
      "bucketAclGrants": [  
        {  
          "grantee": {  
            "id":  
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"  
          },  
          "permission": "READ"  
        }  
      ]  
    }  
  }  
}
```

```
}
```

Saída:

```
{  
  "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"  
}
```

Para obter mais informações, consulte [Visualizando o acesso com o IAM Access Analyzer APIs](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [CreateAccessPreview](#) na Referência de AWS CLI Comandos.

create-analyzer

O código de exemplo a seguir mostra como usar create-analyzer.

AWS CLI

Para criar um analisador

O create-analyzer exemplo a seguir cria um analisador na sua AWS conta.

```
aws accessanalyzer create-analyzer \  
  --analyzer-name example \  
  --type ACCOUNT
```

Saída:

```
{  
  "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"  
}
```

Para obter mais informações, consulte [Introdução às descobertas do AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [CreateAnalyzer](#) na Referência de AWS CLI Comandos.

create-archive-rule

O código de exemplo a seguir mostra como usar create-archive-rule.

AWS CLI

Para criar uma regra de arquivamento para o analisador especificado

O `create-archive-rule` exemplo a seguir cria uma regra de arquivamento para o analisador especificado em sua AWS conta.

```
aws accessanalyzer create-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":  
  ["AWS::IAM::Role"]}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Regras de arquivamento](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [CreateArchiveRule](#) na Referência de AWS CLI Comandos.

delete-analyzer

O código de exemplo a seguir mostra como usar `delete-analyzer`.

AWS CLI

Para excluir o analisador especificado

O `delete-analyzer` exemplo a seguir exclui o analisador especificado em sua AWS conta.

```
aws accessanalyzer delete-analyzer \  
  --analyzer-name example
```

Este comando não produz saída.

Para obter mais informações, consulte [Regras de arquivamento](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteAnalyzer](#) na Referência de AWS CLI Comandos.

delete-archive-rule

O código de exemplo a seguir mostra como usar `delete-archive-rule`.

AWS CLI

Para excluir a regra de arquivamento especificada

O `delete-archive-rule` exemplo a seguir exclui a regra de arquivamento especificada em sua AWS conta.

```
aws accessanalyzer delete-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule
```

Este comando não produz saída.

Para obter mais informações, consulte [Regras de arquivamento](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DeleteArchiveRule](#) na Referência de AWS CLI Comandos.

`get-access-preview`

O código de exemplo a seguir mostra como usar `get-access-preview`.

AWS CLI

Para recuperar informações sobre uma visualização prévia de acesso para o analisador especificado

O `get-access-preview` exemplo a seguir recupera informações sobre uma visualização prévia de acesso para o analisador especificado em sua AWS conta.

```
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
```

Saída:

```
{  
  "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\"Effect\":"Allow\", \"Principal\":{\n\"AWS\":[\n\"arn:aws:iam::111122223333:root\"
],\n\"Action\":[\n\"s3:PutObject\", \"s3:PutObjectAcl\"],\n\"Resource\":\n\"arn:aws:s3:::DOC-
EXAMPLE-BUCKET/*\"}]}}",
          "bucketAclGrants": [
            {
              "permission": "READ",
              "grantee": {
                "id":
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
              }
            }
          ],
          "bucketPublicAccessBlock": {
            "ignorePublicAcls": true,
            "restrictPublicBuckets": true
          }
        }
      },
      "createdAt": "2024-02-17T00:18:44+00:00",
      "status": "COMPLETED"
    }
  }
}

```

Para obter mais informações, consulte [Visualizando o acesso com o IAM Access Analyzer APIs](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [GetAccessPreview](#) na Referência de AWS CLI Comandos.

get-analyzed-resource

O código de exemplo a seguir mostra como usar `get-analyzed-resource`.

AWS CLI

Para recuperar informações sobre um recurso que foi analisado

O `get-analyzed-resource` exemplo a seguir recupera informações sobre um recurso que foi analisado em sua AWS conta.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [GetAnalyzedResource](#) na Referência de AWS CLI Comandos.

get-analyzer

O código de exemplo a seguir mostra como usar `get-analyzer`.

AWS CLI

Para recuperar informações sobre o analisador especificado

O `get-analyzer` exemplo a seguir recupera informações sobre o analisador especificado em sua AWS conta.

```
aws accessanalyzer get-analyzer \  
  --analyzer-name ConsoleAnalyzer-account
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [GetAnalyzer](#) na Referência de AWS CLI Comandos.

get-archive-rule

O código de exemplo a seguir mostra como usar `get-archive-rule`.

AWS CLI

Para recuperar informações sobre uma regra de arquivamento

O `get-archive-rule` exemplo a seguir recupera informações sobre uma regra de arquivamento em sua AWS conta.

```
aws accessanalyzer get-archive-rule \
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
  --rule-name MyArchiveRule
```

Saída:

```
{
  "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"
}
}

```

Para obter mais informações, consulte [Regras de arquivamento](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [GetArchiveRule](#) na Referência de AWS CLI Comandos.

get-finding-v2

O código de exemplo a seguir mostra como usar `get-finding-v2`.

AWS CLI

Para recuperar informações sobre a descoberta especificada

O `get-finding-v2` exemplo a seguir recupera informações sobre a descoberta especificada em sua AWS conta.

```

aws accessanalyzer get-finding-v2 \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
  --id 0910eedb-381e-4e95-adda-0d25c19e6e90

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Analisando as descobertas](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetFindingV2](#) na Referência de AWS CLI Comandos.

get-finding

O código de exemplo a seguir mostra como usar get-finding.

AWS CLI

Para recuperar informações sobre a descoberta especificada

O get-finding exemplo a seguir recupera informações sobre a descoberta especificada em sua AWS conta.

```

aws accessanalyzer get-finding \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \

```

```
--id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Analisando as descobertas](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetFinding](#) na Referência de AWS CLI Comandos.

get-generated-policy

O código de exemplo a seguir mostra como usar `get-generated-policy`.

AWS CLI

Para recuperar a política que foi gerada usando o `StartPolicyGeneration` API

O `get-generated-policy` exemplo a seguir recupera a política que foi gerada usando o `StartPolicyGeneration` API em sua AWS conta.

```
aws accessanalyzer get-generated-policy \
  --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

Saída:

```
{
  "generatedPolicyResult": {
    "generatedPolicies": [
      {
        "policy": "{\n\"Version\":\n\"2012-10-17\", \"Statement\":
[\n\"Sid\":\n\"SupportedServiceSid0\", \"Effect\":\n\"Allow\", \"Action\":
[\n\"access-analyzer:GetAnalyzer\", \"access-analyzer:ListAnalyzers\",
\n\"access-analyzer:ListArchiveRules\", \"access-analyzer:ListFindings
\", \"cloudtrail:DescribeTrails\", \"cloudtrail:GetEventDataStore\",
\n\"cloudtrail:GetEventSelectors\", \"cloudtrail:GetInsightSelectors
\", \"cloudtrail:GetTrailStatus\", \"cloudtrail:ListChannels\",
\n\"cloudtrail:ListEventDataStores\", \"cloudtrail:ListQueries\", \"cloudtrail:ListTags
\", \"cloudtrail:LookupEvents\", \"ec2:DescribeRegions\", \"iam:GetAccountSummary
\", \"iam:GetOpenIDConnectProvider\", \"iam:GetRole\", \"iam:ListAccessKeys\",
\n\"iam:ListAccountAliases\", \"iam:ListOpenIDConnectProviders\", \"iam:ListRoles
\", \"iam:ListSAMLProviders\", \"kms:ListAliases\", \"s3:GetBucketLocation\",
\n\"s3:ListAllMyBuckets\"], \"Resource\":\n\"*\"]}"
      }
    ],
    "properties": {
      "cloudTrailProperties": {
        "endTime": "2024-02-14T22:44:40+00:00",
        "startTime": "2024-02-13T00:30:00+00:00",
        "trailProperties": [
          {
            "allRegions": true,
            "cloudTrailArn": "arn:aws:cloudtrail:us-
west-2:111122223333:trail/my-trail",
            "regions": []
          }
        ]
      },
      "isComplete": false,
      "principalArn": "arn:aws:iam::111122223333:role/Admin"
    }
  },
  "jobDetails": {
    "completedOn": "2024-02-14T22:47:01+00:00",
  }
}
```

```
    "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
    "startedOn": "2024-02-14T22:44:41+00:00",
    "status": "SUCCEEDED"
  }
}
```

Para obter mais informações, consulte [Geração de políticas do IAM Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [GetGeneratedPolicy](#) na Referência de AWS CLI Comandos.

list-access-preview-findings

O código de exemplo a seguir mostra como usar `list-access-preview-findings`.

AWS CLI

Para recuperar uma lista de descobertas de visualização de acesso geradas pela visualização de acesso especificada

O `list-access-preview-findings` exemplo a seguir recupera uma lista de descobertas de visualização de acesso geradas pela visualização prévia de acesso especificada em sua AWS conta.

```
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
```

Saída:

```
{
  "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"
      }
    ]
  }
]
}

```

Para obter mais informações, consulte [Visualizando o acesso com o IAM Access Analyzer APIs](#) no Guia do AWS IAMUsuário.

- Para API obter detalhes, consulte [ListAccessPreviewFindings](#) na Referência de AWS CLI Comandos.

list-access-previews

O código de exemplo a seguir mostra como usar `list-access-previews`.

AWS CLI

Para recuperar uma lista de visualizações de acesso para o analisador especificado

O `list-access-previews` exemplo a seguir recupera uma lista de visualizações de acesso para o analisador especificado em sua conta. AWS

```

aws accessanalyzer list-access-previews \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account

```

Saída:

```

{
  "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"
}
]
```

Para obter mais informações, consulte [Visualizando o acesso com o IAM Access Analyzer APIs](#) no Guia do AWS IAMUsuário.

- Para API obter detalhes, consulte [ListAccessPreviews](#) na Referência de AWS CLI Comandos.

list-analyzed-resources

O código de exemplo a seguir mostra como usar `list-analyzed-resources`.

AWS CLI

Para listar os widgets disponíveis

O `list-analyzed-resources` exemplo a seguir lista os widgets disponíveis na sua AWS conta.

```
aws accessanalyzer list-analyzed-resources \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
  --resource-type AWS::IAM::Role
```

Saída:

```
{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListAnalyzedResources](#) na Referência de AWS CLI Comandos.

list-analyzers

O código de exemplo a seguir mostra como usar `list-analyzers`.

AWS CLI

Para recuperar uma lista de analisadores

O `list-analyzers` exemplo a seguir recupera uma lista de analisadores em sua conta. AWS

```
aws accessanalyzer list-analyzers
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListAnalyzers](#) na Referência de AWS CLI Comandos.

list-archive-rules

O código de exemplo a seguir mostra como usar `list-archive-rules`.

AWS CLI

Para recuperar uma lista de regras de arquivamento criadas para o analisador especificado

O `list-archive-rules` exemplo a seguir recupera uma lista de regras de arquivamento criadas para o analisador especificado em sua AWS conta.

```
aws accessanalyzer list-archive-rules \  
--analyzer-name UnusedAccess-ConsoleAnalyzer-organization
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListArchiveRules](#) na Referência de AWS CLI Comandos.

list-findings-v2

O código de exemplo a seguir mostra como usar `list-findings-v2`.

AWS CLI

Para recuperar uma lista de descobertas geradas pelo analisador especificado

O `list-findings-v2` exemplo a seguir recupera uma lista de descobertas geradas pelo analisador especificado em sua AWS conta. Este exemplo filtra os resultados para incluir somente IAM funções cujo nome contenha `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"]}}'
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListFindingsV2](#) na Referência de AWS CLI Comandos.

list-findings

O código de exemplo a seguir mostra como usar `list-findings`.

AWS CLI

Para recuperar uma lista de descobertas geradas pelo analisador especificado

O `list-findings` exemplo a seguir recupera uma lista de descobertas geradas pelo analisador especificado em sua AWS conta. Este exemplo filtra os resultados para incluir somente IAM funções cujo nome contenha `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"]}}'

```

Saída:

```

{
  "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"
}
]
}
```


Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListFindings](#) na Referência de AWS CLI Comandos.

list-policy-generations

O código de exemplo a seguir mostra como usar `list-policy-generations`.

AWS CLI

Para listar todas as gerações de políticas solicitadas nos últimos sete dias

O `list-policy-generations` exemplo a seguir lista todas as gerações de políticas solicitadas nos últimos sete dias em sua AWS conta.

```
aws accessanalyzer list-policy-generations
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Geração de políticas do IAM Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListPolicyGenerations](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para recuperar uma lista de tags aplicadas ao recurso especificado

O `list-tags-for-resource` exemplo a seguir recupera uma lista de tags aplicadas ao recurso especificado em sua AWS conta.

```
aws accessanalyzer list-tags-for-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

Saída:

```
{  
  "tags": {  
    "Zone-of-trust": "Account",  
    "Name": "ConsoleAnalyzer"  
  }  
}
```

Para obter mais informações, consulte [Geração de políticas do IAM Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

start-policy-generation

O código de exemplo a seguir mostra como usar `start-policy-generation`.

AWS CLI

Para iniciar uma solicitação de geração de política

O `start-policy-generation` exemplo a seguir inicia uma solicitação de geração de política em sua AWS conta.

```
aws accessanalyzer start-policy-generation \
  --policy-generation-details '{"principalArn": "arn:aws:iam::111122223333:role/
Admin"}' \
  --cloud-trail-details file://myfile.json
```

Conteúdo de 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"
    }
  ]
}
```

Saída:

```
{
  "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"
}
```

Para obter mais informações, consulte [Geração de políticas do IAM Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [StartPolicyGeneration](#) na Referência de AWS CLI Comandos.

start-resource-scan

O código de exemplo a seguir mostra como usar start-resource-scan.

AWS CLI

Para iniciar imediatamente uma verificação das políticas aplicadas ao recurso especificado

O start-resource-scan exemplo a seguir inicia imediatamente uma análise das políticas aplicadas ao recurso especificado em sua AWS conta.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Geração de políticas do IAM Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [StartResourceScan](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para adicionar uma tag ao recurso especificado

O tag-resource exemplo a seguir adiciona uma tag ao recurso especificado em sua AWS conta.

```
aws accessanalyzer tag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tags Environment=dev, Purpose=testing
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover tags dos recursos especificados

O `untag-resource` exemplo a seguir remove as tags do recurso especificado em sua AWS conta.

```
aws accessanalyzer untag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tag-keys Environment Purpose
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-archive-rule

O código de exemplo a seguir mostra como usar `update-archive-rule`.

AWS CLI

Para atualizar os critérios e valores da regra de arquivamento especificada

O `update-archive-rule` exemplo a seguir atualiza os critérios e os valores da regra de arquivamento especificada em sua AWS conta.

```
aws accessanalyzer update-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":  
["AWS::IAM::Role"]}}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Regras de arquivamento](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [UpdateArchiveRule](#) na Referência de AWS CLI Comandos.

update-findings

O código de exemplo a seguir mostra como usar `update-findings`.

AWS CLI

Para atualizar o status das descobertas especificadas

O `update-findings` exemplo a seguir atualiza o status das descobertas especificadas em sua AWS conta.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o AWS Identity and Access Management Access Analyzer](#) no Guia do AWS IAM Usuário.

- Para API obter detalhes, consulte [UpdateFindings](#) na Referência de AWS CLI Comandos.

`validate-policy`

O código de exemplo a seguir mostra como usar `validate-policy`.

AWS CLI

Para solicitar a validação de uma política e retornar uma lista de descobertas

O `validate-policy` exemplo a seguir solicita a validação de uma política e retorna uma lista de descobertas. A política no exemplo é uma política de confiança de função para uma função do Amazon Cognito usada para federação de identidade na web. As descobertas geradas a partir da política de confiança estão relacionadas a um valor de `Sid` elemento vazio e a um princípio de política incompatível devido à ação incorreta de assumir a função que está sendo usada, `sts:AssumeRole`. A ação correta de assumir a função para uso com o Cognito é `sts:AssumeRoleWithWebIdentity`

```
aws accessanalyzer validate-policy \  
  --policy-document file://myfile.json \  
  --policy-type RESOURCE_POLICY
```

Conteúdo de `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"
        }
      }
    }
  ]
}
```

Saída:

```
{
  "findings": [
    {
      "findingDetails": "Add a value to the empty string in the Sid element.",
      "findingType": "SUGGESTION",
      "issueCode": "EMPTY_SID_VALUE",
      "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-checks-suggestion-empty-sid-value",
      "locations": [
        {
          "path": [
            {
              "value": "Statement"
            },
            {
              "index": 0
            }
          ]
        }
      ]
    }
  ]
}
```

```

        "value": "Sid"
      }
    ],
    "span": {
      "end": {
        "column": 21,
        "line": 5,
        "offset": 81
      },
      "start": {
        "column": 19,
        "line": 5,
        "offset": 79
      }
    }
  }
]
},
{
  "findingDetails": "The sts:AssumeRole action is invalid with the
following principal(s): cognito-identity.amazonaws.com. Use a SAML provider
principal with the sts:AssumeRoleWithSAML action or use an OIDC provider principal
with the sts:AssumeRoleWithWebIdentity action. Ensure the provider is Federated if
you use either of the two options.",
  "findingType": "ERROR",
  "issueCode": "MISMATCHED_ACTION_FOR_PRINCIPAL",
  "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-mismatched-action-for-principal",
  "locations": [
    {
      "path": [
        {
          "value": "Statement"
        },
        {
          "index": 0
        },
        {
          "value": "Action"
        },
        {
          "index": 0
        }
      ]
    }
  ]
}

```



```
    ],
    "span": {
      "end": {
        "column": 32,
        "line": 11,
        "offset": 274
      },
      "start": {
        "column": 16,
        "line": 11,
        "offset": 258
      }
    }
  },
  {
    "path": [
      {
        "value": "Statement"
      },
      {
        "index": 0
      },
      {
        "value": "Principal"
      },
      {
        "value": "Federated"
      }
    ],
    "span": {
      "end": {
        "column": 61,
        "line": 8,
        "offset": 202
      },
      "start": {
        "column": 29,
        "line": 8,
        "offset": 170
      }
    }
  }
]
},
```

```
{
  "findingDetails": "The following actions: sts:TagSession are not
supported by the condition key cognito-identity.amazonaws.com:aud. The condition
will not be evaluated for these actions. We recommend that you move these actions
to a different statement without this condition key.",
  "findingType": "ERROR",
  "issueCode": "UNSUPPORTED_ACTION_FOR_CONDITION_KEY",
  "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/
access-analyzer-reference-policy-checks.html#access-analyzer-reference-policy-
checks-error-unsupported-action-for-condition-key",
  "locations": [
    {
      "path": [
        {
          "value": "Statement"
        },
        {
          "index": 0
        },
        {
          "value": "Action"
        },
        {
          "index": 1
        }
      ],
      "span": {
        "end": {
          "column": 32,
          "line": 12,
          "offset": 308
        },
        "start": {
          "column": 16,
          "line": 12,
          "offset": 292
        }
      }
    },
    {
      "path": [
        {
          "value": "Statement"
        },

```

```
    {
      "index": 0
    },
    {
      "value": "Condition"
    },
    {
      "value": "StringEquals"
    },
    {
      "value": "cognito-identity.amazonaws.com:aud"
    }
  ],
  "span": {
    "end": {
      "column": 79,
      "line": 16,
      "offset": 464
    },
    "start": {
      "column": 58,
      "line": 16,
      "offset": 443
    }
  }
}
]
}
]
```

Para obter mais informações, consulte [Verificações de validação de políticas](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [ValidatePolicy](#) na Referência de AWS CLI Comandos.

Exemplos do Image Builder usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Image Builder.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-component

O código de exemplo a seguir mostra como usar `create-component`.

AWS CLI

Para criar um componente

O `create-component` exemplo a seguir cria um componente que usa um arquivo de JSON documento e faz referência a um documento de componente no YAML formato que é carregado em um bucket do Amazon S3.

```
aws imagebuilder create-component \  
  --cli-input-json file://create-component.json
```

Conteúdo de `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"  
}
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [CreateComponent](#) na Referência de AWS CLI Comandos.

create-distribution-configuration

O código de exemplo a seguir mostra como usar create-distribution-configuration.

AWS CLI

Para criar uma configuração de distribuição

O create-distribution-configuration exemplo a seguir cria uma configuração de distribuição usando um JSON arquivo.

```
aws imagebuilder create-distribution-configuration \
  --cli-input-json file:/create-distribution-configuration.json
```

Conteúdo de 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"
        ]
      }
    }
  }
]
}

```

Saída:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexempleredistribution"
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [CreateDistributionConfiguration](#) na Referência de AWS CLI Comandos.

create-image-pipeline

O código de exemplo a seguir mostra como usar create-image-pipeline.

AWS CLI

Para criar um pipeline de imagens

O `create-image-pipeline` exemplo a seguir cria um pipeline de imagens usando um JSON arquivo.

```
aws imagebuilder create-image-pipeline \  
--cli-input-json file://create-image-pipeline.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [CreateImagePipeline](#) na Referência de AWS CLI Comandos.

create-image-recipe

O código de exemplo a seguir mostra como usar `create-image-recipe`.

AWS CLI

Para criar uma receita

O `create-image-recipe` exemplo a seguir cria uma receita de imagem usando um JSON arquivo. Os componentes são instalados na ordem em que são especificados.

```
aws imagebuilder create-image-recipe \  
  --cli-input-json file://create-image-recipe.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "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"
```



```
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [CreateImageRecipe](#) na Referência de AWS CLI Comandos.

create-image

O código de exemplo a seguir mostra como usar `create-image`.

AWS CLI

Para criar uma imagem

O `create-image` exemplo a seguir cria uma imagem.

```
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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [CreateImage](#) na Referência de AWS CLI Comandos.

create-infrastructure-configuration

O código de exemplo a seguir mostra como usar `create-infrastructure-configuration`.

AWS CLI

Para criar uma configuração de infraestrutura

O `create-infrastructure-configuration` exemplo a seguir cria uma configuração de infraestrutura usando um JSON arquivo.

```
aws imagebuilder create-infrastructure-configuration \  
  --cli-input-json file://create-infrastructure-configuration.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [CreateInfrastructureConfiguration](#) na Referência de AWS CLI Comandos.

delete-component

O código de exemplo a seguir mostra como usar delete-component.

AWS CLI

Para excluir um componente

O delete-component exemplo a seguir exclui uma versão de compilação do componente especificando sua. ARN

```
aws imagebuilder delete-component \  
  --component-build-version-arn arn:aws:imagebuilder:us-  
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [DeleteComponent](#) na Referência de AWS CLI Comandos.

delete-image-pipeline

O código de exemplo a seguir mostra como usar delete-image-pipeline.

AWS CLI

Para excluir um pipeline de imagens

O `delete-image-pipeline` exemplo a seguir exclui um pipeline de imagem especificando seu ARN

```
aws imagebuilder delete-image-pipeline \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
my-example-pipeline
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [DeletImagePipeline](#) na Referência de AWS CLI Comandos.

delete-image-recipe

O código de exemplo a seguir mostra como usar `delete-image-recipe`.

AWS CLI

Para excluir uma receita de imagem

O `delete-image-recipe` exemplo a seguir exclui uma receita de imagem especificando sua ARN

```
aws imagebuilder delete-image-recipe \  
  --image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/  
mybasicrecipe/2019.12.03
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
mybasicrecipe/2019.12.03"
```

```
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [DeleteImageRecipe](#) na Referência de AWS CLI Comandos.

delete-image

O código de exemplo a seguir mostra como usar delete-image.

AWS CLI

Para excluir uma imagem

O delete-image exemplo a seguir exclui uma versão de criação de imagem especificando sua ARN

```
aws imagebuilder delete-image \
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-
  example-image/2019.12.02/1
```

Saída:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
  mybasicreceipe/2019.12.03/1"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [DeleteImage](#) na Referência de AWS CLI Comandos.

delete-infrastructure-configuration

O código de exemplo a seguir mostra como usar delete-infrastructure-configuration.

AWS CLI

Para excluir uma configuração de infraestrutura

O `delete-infrastructure-configuration` exemplo a seguir exclui um pipeline de imagem especificando seu ARN

```
aws imagebuilder delete-infrastructure-configuration \  
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-  
east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [DeleteInfrastructureConfiguration](#) na Referência de AWS CLI Comandos.

get-component-policy

O código de exemplo a seguir mostra como usar `get-component-policy`.

AWS CLI

Para obter detalhes da política de componentes

O `get-component-policy` exemplo a seguir lista os detalhes de uma política de componentes especificando sua ARN.

```
aws imagebuilder get-component-policy \  
  --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-  
example-component/2019.12.03/1
```

Saída:

```
{  
  "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" ] } ] }"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de imagem do EC2 Image Builder usando o AWS CLI](https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html) < <https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html> > __ no Guia do usuário do Image Builder. EC2

- Para API obter detalhes, consulte [GetComponentPolicy](#) na Referência de AWS CLI Comandos.

get-component

O código de exemplo a seguir mostra como usar `get-component`.

AWS CLI

Para obter detalhes do componente

O `get-component` exemplo a seguir lista os detalhes de um componente especificando seu ARN.

```
aws imagebuilder get-component \
  --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/component-name/1.0.0/1
```

Saída:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "component": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/component-
name/1.0.0/1",
    "name": "component-name",
    "version": "1.0.0",
    "type": "TEST",
    "platform": "Linux",
    "owner": "123456789012",
    "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
testing document.\nschemaVersion: 1.0\n\nphases:\n - name: test\n   steps:\n
- name: HelloWorldStep\n       action: ExecuteBash\n       inputs:\n
commands:\n       - echo \"Hello World! Test.\"\n",
```

```

    "encrypted": true,
    "dateCreated": "2020-01-27T20:43:30.306Z",
    "tags": {}
  }
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetComponentna](#) Referência de AWS CLI Comandos.

get-distribution-configuration

O código de exemplo a seguir mostra como usar `get-distribution-configuration`.

AWS CLI

Para obter os detalhes de uma configuração de distribuição

O `get-distribution-configuration` exemplo a seguir exibe os detalhes de uma configuração de distribuição especificando sua ARN.

```

aws imagebuilder get-distribution-configuration \
  --distribution-configuration-arn arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution

```

Saída:

```

{
  "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": {}
}
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetDistributionConfiguration](#) na Referência de AWS CLI Comandos.

get-image-pipeline

O código de exemplo a seguir mostra como usar `get-image-pipeline`.

AWS CLI

Para obter detalhes do pipeline de imagens

O `get-image-pipeline` exemplo a seguir lista os detalhes de um pipeline de imagem especificando seu ARN.

```
aws imagebuilder get-image-pipeline \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetImagePipeline](#) na Referência de AWS CLI Comandos.

get-image-policy

O código de exemplo a seguir mostra como usar `get-image-policy`.

AWS CLI

Para obter detalhes da política de imagem

O `get-image-policy` exemplo a seguir lista os detalhes de uma política de imagem especificando sua ARN.

```
aws imagebuilder get-image-policy \  
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-  
image/2019.12.03/1
```

Saída:

```
{  
  "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\" ] } ] }"
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetImagePolicy](#) na Referência de AWS CLI Comandos.

get-image-recipe-policy

O código de exemplo a seguir mostra como usar `get-image-recipe-policy`.

AWS CLI

Para obter detalhes da política de receitas de imagens

O `get-image-recipe-policy` exemplo a seguir lista os detalhes de uma política de receitas de imagens especificando sua ARN.

```
aws imagebuilder get-image-recipe-policy \  
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:recipe/my-example-recipe/2019.12.03/1
```

```
--image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-recipe/2019.12.03/1
```

Saída:

```
{
  "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\" ] } ] }"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetImageRecipePolicy](#) na Referência de AWS CLI Comandos.

get-image

O código de exemplo a seguir mostra como usar `get-image`.

AWS CLI

Para obter detalhes da imagem

O `get-image` exemplo a seguir lista os detalhes de uma imagem especificando sua ARN.

```
aws imagebuilder get-image \
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1
```

Saída:

```
{
  "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": {}
}
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetImage](#) na Referência de AWS CLI Comandos.

get-infrastructure-configuration

O código de exemplo a seguir mostra como usar `get-infrastructure-configuration`.

AWS CLI

Para obter detalhes da configuração da infraestrutura

O `get-infrastructure-configuration` exemplo a seguir lista os detalhes de uma configuração de infraestrutura especificando sua ARN.

```

aws imagebuilder get-infrastructure-configuration \
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure

```

Saída:

```
{
  "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": {}
  }
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [GetInfrastructureConfiguration](#) na Referência de AWS CLI Comandos.

import-component

O código de exemplo a seguir mostra como usar `import-component`.

AWS CLI

Para importar um componente

O `import-component` exemplo a seguir importa um script preexistente usando um JSON arquivo.

```
aws imagebuilder import-component \  
  --cli-input-json file://import-component.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ImportComponent](#) na Referência de AWS CLI Comandos.

list-component-build-versions

O código de exemplo a seguir mostra como usar `list-component-build-versions`.

AWS CLI

Para listar as versões de compilação de componentes

O `list-component-build-versions` exemplo a seguir lista as versões de construção do componente com uma versão semântica específica.

```
aws imagebuilder list-component-build-versions --component-  
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/  
myexamplecomponent/2019.12.02
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListComponentBuildVersions](#) na Referência de AWS CLI Comandos.

list-components

O código de exemplo a seguir mostra como usar `list-components`.

AWS CLI

Para listar todas as versões semânticas do componente

O `list-components` exemplo a seguir lista todas as versões semânticas do componente às quais você tem acesso. Opcionalmente, você pode filtrar se deseja listar componentes de sua propriedade, da Amazon ou que foram compartilhados com você por outras contas.

```
aws imagebuilder list-components
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListComponents](#) na Referência de AWS CLI Comandos.

list-distribution-configurations

O código de exemplo a seguir mostra como usar `list-distribution-configurations`.

AWS CLI

Para listar distribuições

O `list-distribution-configurations` exemplo a seguir lista todas as suas distribuições.

```
aws imagebuilder list-distribution-configurations
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListDistributionConfigurations](#) na Referência de AWS CLI Comandos.

list-image-build-versions

O código de exemplo a seguir mostra como usar `list-image-build-versions`.

AWS CLI

Para listar as versões de criação de imagens

O `list-image-build-versions` exemplo a seguir lista todas as versões de criação de imagem com uma versão semântica.

```
aws imagebuilder list-image-build-versions \
  --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03
```

Saída:

```
{
  "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\":
\n\"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": {}
    }
  ]
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListImageBuildVersions](#) na Referência de AWS CLI Comandos.

list-image-pipeline-images

O código de exemplo a seguir mostra como usar `list-image-pipeline-images`.

AWS CLI

Para listar imagens do pipeline de imagens

O `list-image-pipeline-images` exemplo a seguir lista todas as imagens que foram criadas por um pipeline de imagens específico.

```
aws imagebuilder list-image-pipeline-images \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

Saída:

```
{  
  "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"
    }
  }
]
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListImagePipelineImages](#) na Referência de AWS CLI Comandos.

list-image-recipes

O código de exemplo a seguir mostra como usar `list-image-recipes`.

AWS CLI

Para listar receitas de imagens

O `list-image-recipes` exemplo a seguir lista todas as suas receitas de imagens.

```
aws imagebuilder list-image-recipes
```

Saída:

```
{
  "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"
      }
    }
  ]
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListImageRecipes](#) na Referência de AWS CLI Comandos.

list-images

O código de exemplo a seguir mostra como usar `list-images`.

AWS CLI

Para listar imagens

O `list-images` exemplo a seguir lista todas as versões semânticas às quais você tem acesso.

```
aws imagebuilder list-images
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListImages](#) na Referência de AWS CLI Comandos.

list-infrastructure-configurations

O código de exemplo a seguir mostra como usar `list-infrastructure-configurations`.

AWS CLI

Para listar as configurações de infraestrutura

O `list-infrastructure-configurations` exemplo a seguir lista todas as suas configurações de infraestrutura.

```
aws imagebuilder list-infrastructure-configurations
```

Saída:

```
{
  "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"  
    }  
  }  
]  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListInfrastructureConfigurations](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um recurso específico

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um recurso específico.

```
aws imagebuilder list-tags-for-resource \  
  --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

Saída:

```
{
  "tags": {
    "KeyName": "KeyValue"
  }
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-component-policy

O código de exemplo a seguir mostra como usar `put-component-policy`.

AWS CLI

Para aplicar uma política de recursos a um componente

O `put-component-policy` comando a seguir aplica uma política de recursos a um componente de compilação para permitir o compartilhamento entre contas de componentes de compilação. Recomendamos que você use o RAM CLI comando `create-resource-share`. Se você usar o CLI comando EC2 Image Builder `put-component-policy`, também deverá usar o RAM CLI comando `promote-resource-share-create-from-policy` para que o recurso fique visível para todos os diretores com quem o recurso é compartilhado.

```
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" ] } ] }'
```

Saída:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/examplecomponent/2019.12.02/1"
```

```
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [PutComponentPolicy](#) na Referência de AWS CLI Comandos.

put-image-policy

O código de exemplo a seguir mostra como usar `put-image-policy`.

AWS CLI

Para aplicar uma política de recursos a uma imagem

O `put-image-policy` comando a seguir aplica uma política de recursos a uma imagem para permitir o compartilhamento de imagens entre contas. Recomendamos que você use o RAM CLI comando `create-resource-share`. Se você usar o CLI comando EC2 Image Builder `put-image-policy`, também deverá usar o RAM CLI comando `promote-resource-share-create -from-policy` para que o recurso fique visível para todos os diretores com quem o recurso é compartilhado.

```
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" ] } ] }'
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-  
image/2019.12.02/1"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [PutImagePolicy](#) na Referência de AWS CLI Comandos.

put-image-recipe-policy

O código de exemplo a seguir mostra como usar `put-image-recipe-policy`.

AWS CLI

Para aplicar uma política de recursos a uma receita de imagem

O `put-image-recipe-policy` comando a seguir aplica uma política de recursos a uma receita de imagem para permitir o compartilhamento entre contas de receitas de imagens. Recomendamos que você use o RAM CLI comando `create-resource-share`. Se você usar o CLI comando EC2 Image Builder `put-image-recipe-policy`, também deverá usar o RAM CLI comando `promote-resource-share-create-from-policy` para que o recurso fique visível para todos os diretores com quem o recurso é compartilhado.

```
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" ] } ] }'
```

Saída:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
  example-image-recipe/2019.12.02/1"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [PutImageRecipePolicy](#) na Referência de AWS CLI Comandos.

start-image-pipeline-execution

O código de exemplo a seguir mostra como usar `start-image-pipeline-execution`.

AWS CLI

Para iniciar um pipeline de imagens manualmente

O `start-image-pipeline-execution` exemplo a seguir inicia manualmente um pipeline de imagens.

```
aws imagebuilder start-image-pipeline-execution \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
  mywindows2016pipeline
```

Saída:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/  
  mybasicreceipe/2019.12.03/1"  
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [StartImagePipelineExecution](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

O `tag-resource` exemplo a seguir adiciona e marca um recurso no EC2 Image Builder usando um JSON arquivo.

```
aws imagebuilder tag-resource \  
  --cli-input-json file://tag-resource.json
```

Conteúdo de `tag-resource.json`:


```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
  "tags": {
    "KeyName": "KeyValue"
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove uma tag de um recurso usando um JSON arquivo.

```
aws imagebuilder untag-resource \
  --cli-input-json file://tag-resource.json
```

Conteúdo de `untag-resource.json`:

```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
  "tagKeys": [
    "KeyName"
  ]
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-distribution-configuration

O código de exemplo a seguir mostra como usar update-distribution-configuration.

AWS CLI

Para atualizar uma configuração de distribuição

O update-distribution-configuration exemplo a seguir atualiza uma configuração de distribuição usando um JSON arquivo.

```
aws imagebuilder update-distribution-configuration \  
  --cli-input-json file://update-distribution-configuration.json
```

Conteúdo de update-distribution-configuration.json:

```
{  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-  
west-2:123456789012:distribution-configuration/myexampledistribution",  
  "description": "Copies AMI to eu-west-2 and exports to S3",  
  "distributions": [  
    {  
      "region": "us-west-2",  
      "amiDistributionConfiguration": {  
        "name": "Name {{imagebuilder:buildDate}}",  
        "description": "An example image name with parameter references"  
      }  
    },  
    {  
      "region": "eu-west-2",  
      "amiDistributionConfiguration": {  
        "name": "My {{imagebuilder:buildVersion}} image  
{{imagebuilder:buildDate}}"  
      }  
    }  
  ]  
}
```

Saída:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [UpdateDistributionConfiguration](#) na Referência de AWS CLI Comandos.

update-image-pipeline

O código de exemplo a seguir mostra como usar update-image-pipeline.

AWS CLI

Para atualizar um pipeline de imagens

O update-image-pipeline exemplo a seguir atualiza um pipeline de imagem usando um JSON arquivo.

```
aws imagebuilder update-image-pipeline \
  --cli-input-json file://update-image-pipeline.json
```

Conteúdo de 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"
}

```

Saída:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}

```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [UpdateImagePipeline](#) na Referência de AWS CLI Comandos.

update-infrastructure-configuration

O código de exemplo a seguir mostra como usar update-infrastructure-configuration.

AWS CLI

Para atualizar uma configuração de infraestrutura

O update-infrastructure-configuration exemplo a seguir atualiza uma configuração de infraestrutura usando um JSON arquivo.

```

aws imagebuilder update-infrastructure-configuration \
  --cli-input-json file:/update-infrastructure-configuration.json

```

Conteúdo de 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"
}
```

Saída:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Configurando e gerenciando um pipeline de EC2 imagem do Image Builder usando o AWS CLI](#) no Guia do EC2 Usuário do Image Builder.

- Para API obter detalhes, consulte [UpdateInfrastructureConfiguration](#) na Referência de AWS CLI Comandos.

Exemplos do Incident Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Incident Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-replication-set

O código de exemplo a seguir mostra como usar `create-replication-set`.

AWS CLI

Para criar o conjunto de replicação

O `create-replication-set` exemplo a seguir cria o conjunto de replicação que o Incident Manager usa para replicar e criptografar dados em sua conta da Amazon Web Services. Este exemplo usa as regiões `us-east-1` e `us-east-2` ao criar o conjunto de replicação.

```
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"}}'
```

Saída:

```
{  
  "replicationSetArns": [  
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"  
  ]  
}
```

Para obter mais informações, consulte [Usando o conjunto de replicação do Incident Manager](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [CreateReplicationSet](#) na Referência de AWS CLI Comandos.

create-response-plan

O código de exemplo a seguir mostra como usar `create-response-plan`.

AWS CLI

Para criar um plano de resposta

O `create-response-plan` exemplo a seguir cria um plano de resposta com os detalhes especificados.

```
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"}]
```

Saída:

```
{
  "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}
```

Para obter mais informações, consulte [Preparação de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [CreateResponsePlan](#) na Referência de AWS CLI Comandos.

create-timeline-event

O código de exemplo a seguir mostra como usar `create-timeline-event`.

AWS CLI

Exemplo 1: Para criar um evento de cronograma personalizado

O `create-timeline-event` exemplo a seguir cria um evento de cronograma personalizado no horário especificado no incidente especificado.

```
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"
```

Saída:

```
{  
  "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",  
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"  
}
```

Exemplo 2: Para criar um evento de cronograma com uma nota de incidente

O `create-timeline-event` exemplo a seguir cria um evento de cronograma listado no painel “Notas do 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"} ]'
```

Saída:

```
{  
  "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",  
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"  
}
```

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [CreateTimelineEvent](#) na Referência de AWS CLI Comandos.

delete-incident-record

O código de exemplo a seguir mostra como usar `delete-incident-record`.

AWS CLI

Para excluir um registro de incidente

O `delete-incident-record` exemplo a seguir exclui o registro de incidente especificado.

```
aws ssm-incidents delete-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Este comando não produz saída.

Para obter mais informações, consulte [Rastreamento de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteIncidentRecord](#) na Referência de AWS CLI Comandos.

`delete-replication-set`

O código de exemplo a seguir mostra como usar `delete-replication-set`.

AWS CLI

Para excluir o conjunto de replicação

O `delete-replication-set` exemplo a seguir exclui o conjunto de replicação da sua conta da Amazon Web Services. A exclusão do conjunto de replicação também exclui todos os dados do Incident Manager. Isso não pode ser desfeito.

```
aws ssm-incidents delete-replication-set \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-bb3f-413c-08df53673b57"
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o conjunto de replicação do Incident Manager](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteReplicationSet](#) na Referência de AWS CLI Comandos.

`delete-resource-policy`

O código de exemplo a seguir mostra como usar `delete-resource-policy`.

AWS CLI

Para excluir uma política de recursos

O `delete-resource-policy` exemplo a seguir exclui uma política de recursos de um plano de resposta. Isso revogará o acesso do diretor ou da organização com a qual o plano de resposta foi compartilhado.

```
aws ssm-incidents delete-resource-policy \  
  --policy-id "be8b57191f0371f1c6827341aa3f0a03" \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com contatos compartilhados e planos de resposta](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteResourcePolicy](#) na Referência de AWS CLI Comandos.

`delete-response-plan`

O código de exemplo a seguir mostra como usar `delete-response-plan`.

AWS CLI

Para excluir um plano de resposta

O `delete-response-plan` exemplo a seguir exclui o plano de resposta especificado.

```
aws ssm-incidents delete-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

Este comando não produz saída.

Para obter mais informações, consulte [Preparação de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteResponsePlan](#) na Referência de AWS CLI Comandos.

`delete-timeline-event`

O código de exemplo a seguir mostra como usar `delete-timeline-event`.

AWS CLI

Para excluir um evento da linha do tempo

O `delete-timeline-event` exemplo a seguir exclui um evento de cronograma personalizado do registro de incidente especificado.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteTimelineEvent](#) na Referência de AWS CLI Comandos.

get-incident-record

O código de exemplo a seguir mostra como usar `get-incident-record`.

AWS CLI

Para obter um registro de incidentes

O `get-incident-record` exemplo a seguir obtém detalhes sobre o registro de incidente especificado.

```
aws ssm-incidents get-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-  
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetIncidentRecord](#) na Referência de AWS CLI Comandos.

get-replication-set

O código de exemplo a seguir mostra como usar get-replication-set.

AWS CLI

Para obter o conjunto de replicação

O get-replication-set exemplo a seguir obtém os detalhes do conjunto de replicação que o Incident Manager usa para replicar e criptografar dados em sua conta da Amazon Web Services.

```

aws ssm-incidents get-replication-set \
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Usando o conjunto de replicação do Incident Manager](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetReplicationSet](#) na Referência de AWS CLI Comandos.

get-resource-policies

O código de exemplo a seguir mostra como usar `get-resource-policies`.

AWS CLI

Para listar políticas de recursos para um plano de resposta

O `command-name` exemplo a seguir lista as políticas de recursos associadas ao plano de resposta especificado.

```

aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"

```

Saída:

```
{
```

```

"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"
  }
]
}

```

Para obter mais informações, consulte Como [trabalhar com contatos compartilhados e planos de resposta](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetResourcePolicies](#) na Referência de AWS CLI Comandos.

get-response-plan

O código de exemplo a seguir mostra como usar `get-response-plan`.

AWS CLI

Para obter detalhes de um plano de resposta

O command-name exemplo a seguir mostra detalhes sobre um plano de resposta específico em sua AWS conta.

```

aws ssm-incidents get-response-plan \
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"

```

Saída:

```

{
  "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"
}
```

Para obter mais informações, consulte [Preparação de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetResponsePlan](#) na Referência de AWS CLI Comandos.

get-timeline-event

O código de exemplo a seguir mostra como usar `get-timeline-event`.

AWS CLI

Para obter detalhes de um evento cronológico

O `get-timeline-event` exemplo a seguir retorna detalhes do evento de cronograma especificado.

```
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"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetTimelineEvent](#) na Referência de AWS CLI Comandos.

list-incident-records

O código de exemplo a seguir mostra como usar `list-incident-records`.

AWS CLI

Para listar registros de incidentes

O `command-name` exemplo a seguir lista os registros de incidentes em sua conta da Amazon Web Services.

```
aws ssm-incidents list-incident-records
```

Saída:

```
{
```



```

    "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Lista de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListIncidentRecords](#) na Referência de AWS CLI Comandos.

list-related-items

O código de exemplo a seguir mostra como usar `list-related-items`.

AWS CLI

Para listar itens relacionados

O `list-related-items` exemplo a seguir lista os itens relacionados ao incidente especificado.

```

aws ssm-incidents list-related-items \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"

```

Saída:

```

{
  "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"
}
]
}

```

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListRelatedItems](#) na Referência de AWS CLI Comandos.

list-replication-sets

O código de exemplo a seguir mostra como usar `list-replication-sets`.

AWS CLI

Para listar o conjunto de replicação

O `list-replication-set` exemplo a seguir lista o conjunto de replicação que o Incident Manager usa para replicar e criptografar dados em sua conta. AWS

```
aws ssm-incidents list-replication-sets
```

Saída:

```
{
```

```
"replicationSetArns": [  
  "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"  
]  
}
```

Para obter mais informações, consulte [Usando o conjunto de replicação do Incident Manager](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListReplicationSets](#) na Referência de AWS CLI Comandos.

list-response-plans

O código de exemplo a seguir mostra como usar `list-response-plans`.

AWS CLI

Para listar os planos de resposta disponíveis

O `list-response-plans` exemplo a seguir lista os planos de resposta disponíveis em sua conta da Amazon Web Services.

```
aws ssm-incidents list-response-plans
```

Saída:

```
{  
  "responsePlanSummaries": [  
    {  
      "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan",  
      "displayName": "Example response plan",  
      "name": "Example-Response-Plan"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Preparação de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListResponsePlans](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um plano de resposta

O `list-tags-for-resource` exemplo a seguir lista as tags associadas ao plano de resposta especificado.

```
aws ssm-incidents list-tags-for-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan"
```

Saída:

```
{  
  "tags": {  
    "group1": "1"  
  }  
}
```

Para obter mais informações, consulte [Marcação](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-timeline-events

O código de exemplo a seguir mostra como usar `list-timeline-events`.

AWS CLI

Para listar eventos cronológicos de um incidente

O `command-name` exemplo a seguir lista os eventos da linha do tempo do incidente especificado.

```
aws ssm-incidents list-timeline-events \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListTimelineEvents](#) na Referência de AWS CLI Comandos.

put-resource-policy

O código de exemplo a seguir mostra como usar `put-resource-policy`.

AWS CLI

Para compartilhar um plano de resposta e incidentes

O `command-name` exemplo a seguir adiciona uma política de recursos ao `Example-Response-Plan` que compartilha o plano de resposta e os incidentes associados com o principal especificado.

```

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/*\"]}]}"

```

Saída:

```
{
  "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

Para obter mais informações, consulte Como [trabalhar com contatos compartilhados e planos de resposta](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [PutResourcePolicy](#) na Referência de AWS CLI Comandos.

start-incident

O código de exemplo a seguir mostra como usar `start-incident`.

AWS CLI

Para iniciar um incidente

O `start-incident` exemplo a seguir inicia um incidente usando o plano de resposta especificado.

```
aws ssm-incidents start-incident \
  --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Saída:

```
{
  "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
}
```

Para obter mais informações, consulte [Criação de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [StartIncident](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um plano de resposta

O `tag-resource` exemplo a seguir marca um plano de resposta especificado com o par chave-valor da tag fornecido.

```
aws ssm-incidents tag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \  
  --tags '{"group1":"1"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um plano de resposta

O `untag-resource` exemplo a seguir remove as tags especificadas do plano de resposta.

```
aws ssm-incidents untag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \  
  --tag-keys ['group1']'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-deletion-protection

O código de exemplo a seguir mostra como usar `update-deletion-protection`.

AWS CLI

Para atualizar a proteção contra exclusão do conjunto de replicação

O `update-deletion-protection` exemplo a seguir atualiza a proteção contra exclusão em sua conta para evitar que você exclua a última região em seu conjunto de replicação.

```
aws ssm-incidents update-deletion-protection \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/  
a2bcc5c9-0f53-8047-7fef-c20749989b40" \  
  --deletion-protected
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o conjunto de replicação do Incident Manager](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateDeletionProtection](#) na Referência de AWS CLI Comandos.

update-incident-record

O código de exemplo a seguir mostra como usar `update-incident-record`.

AWS CLI

Para atualizar um registro de incidente

O `command-name` exemplo a seguir resolve o incidente especificado.

```
aws ssm-incidents update-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-  
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \  
  --status "RESOLVED"
```

Este comando não produz saída.

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateIncidentRecord](#) na Referência de AWS CLI Comandos.

update-related-items

O código de exemplo a seguir mostra como usar `update-related-items`.

AWS CLI

Para atualizar um item relacionado a incidentes

O `update-related-item` exemplo a seguir remove um item relacionado do registro de incidente especificado.

```
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"}}}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateRelatedItems](#) na Referência de AWS CLI Comandos.

update-replication-set

O código de exemplo a seguir mostra como usar `update-replication-set`.

AWS CLI

Para atualizar um conjunto de replicação

O `command-name` exemplo a seguir exclui a região `us-east-2` do conjunto de replicação.

```
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"}}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o conjunto de replicação do Incident Manager](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateReplicationSet](#) na Referência de AWS CLI Comandos.

update-response-plan

O código de exemplo a seguir mostra como usar `update-response-plan`.

AWS CLI

Para atualizar um plano de resposta

O `update-response-plan` exemplo a seguir remove um canal de bate-papo do plano de resposta especificado.

```
aws ssm-incidents update-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \  
  --chat-channel '{"empty":{}}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Preparação de incidentes](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateResponsePlan](#) na Referência de AWS CLI Comandos.

update-timeline-event

O código de exemplo a seguir mostra como usar `update-timeline-event`.

AWS CLI

Para atualizar um evento na linha do tempo

O `update-timeline-event` exemplo a seguir atualiza a hora em que o evento ocorreu.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Detalhes do incidente](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateTimelineEvent](#) na Referência de AWS CLI Comandos.

Exemplos de contatos do Incident Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com os contatos do Incident Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-page

O código de exemplo a seguir mostra como usar `accept-page`.

AWS CLI

Para aceitar uma página durante um engajamento

O `accept-page` exemplo a seguir usa um código de aceitação enviado ao canal de contato para aceitar uma página.

```
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
```

Esse comando não produz nenhuma saída

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [AcceptPage](#) na Referência de AWS CLI Comandos.

activate-contact-channel

O código de exemplo a seguir mostra como usar `activate-contact-channel`.

AWS CLI

Ativar o canal de contato de um contato

O `activate-contact-channel` exemplo a seguir ativa um canal de contato e o torna utilizável como parte de um 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" \  
  --activation-code "466136"
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ActivateContactChannel](#) na Referência de AWS CLI Comandos.

command-name

O código de exemplo a seguir mostra como usar `command-name`.

AWS CLI

Para excluir um contato

O `command-name` exemplo a seguir exclui um contato. O contato não estará mais acessível a partir de nenhum plano de escalonamento que se refira a ele.

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [CommandName](#) na Referência de AWS CLI Comandos.

create-contact-channel

O código de exemplo a seguir mostra como usar `create-contact-channel`.

AWS CLI

Para criar um canal de contato

Cria um canal de contato do tipo SMS para o contato Akua Mansa. Os canais de contato podem ser criados do tipo SMS, EMAIL, ou VOICE.

```
aws ssm-contacts create-contact-channel \
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
  --name "akuas sms-test" \
  --type SMS \
  --delivery-address '{"SimpleAddress": "+15005550199"}'
```

Saída:

```
{
  "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [CreateContactChannel](#) na Referência de AWS CLI Comandos.

create-contact

O código de exemplo a seguir mostra como usar `create-contact`.

AWS CLI

Para criar um contato

O `create-contact` exemplo a seguir cria um contato em seu ambiente com um plano em branco. O plano pode ser atualizado após a criação dos canais de contato. Use o `create-contact-channel` comando com a saída ARN desse comando. Depois de criar canais de contato para esse contato, use `update-contact` para atualizar o plano.

```
aws ssm-contacts create-contact \  
  --alias "akuam" \  
  --display-name "Akua Mansa" \  
  --type PERSONAL \  
  --plan '{"Stages": []}'
```

Saída:

```
{  
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [CreateContact](#) na Referência de AWS CLI Comandos.

`deactivate-contact-channel`

O código de exemplo a seguir mostra como usar `deactivate-contact-channel`.

AWS CLI

Para desativar um canal de contato

O `deactivate-contact-channel` exemplo a seguir desativa um canal de contato. Desativar um canal de contato significa que o canal de contato não será mais paginado durante um incidente. Você também pode reativar um canal de contato a qualquer momento usando o `activate-contact-channel` comando.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeactivateContactChannel](#) na Referência de AWS CLI Comandos.

delete-contact-channel

O código de exemplo a seguir mostra como usar delete-contact-channel.

AWS CLI

Para excluir um canal de contato

O delete-contact-channel exemplo a seguir exclui um canal de contato. A exclusão de um canal de contato garante que o canal de contato não seja paginado durante um 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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteContactChannel](#) na Referência de AWS CLI Comandos.

delete-contact

O código de exemplo a seguir mostra como usar delete-contact.

AWS CLI

Para excluir um contato

O delete-contact exemplo a seguir exclui um contato. O contato não estará mais acessível a partir de nenhum plano de escalonamento que se refira a ele.

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DeleteContact](#) na Referência de AWS CLI Comandos.

describe-engagement

O código de exemplo a seguir mostra como usar `describe-engagement`.

AWS CLI

Para descrever os detalhes de um compromisso

O `describe-engagement` exemplo a seguir lista os detalhes de um compromisso com um plano de contato ou escalonamento. O assunto e o conteúdo são enviados para os canais de contato.

```
aws ssm-contacts describe-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DescribeEngagement](#) na Referência de AWS CLI Comandos.

describe-page

O código de exemplo a seguir mostra como usar `describe-page`.

AWS CLI

Para listar os detalhes de uma página em um canal de contato

O `describe-page` exemplo a seguir lista os detalhes de uma página em um canal de contato. A página incluirá o assunto e o conteúdo fornecidos.

```
aws ssm-contacts describe-page \  
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-  
e606-498a-861b-25726292eb93"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [DescribePage](#) na Referência de AWS CLI Comandos.

`get-contact-channel`

O código de exemplo a seguir mostra como usar `get-contact-channel`.

AWS CLI

Para listar os detalhes de um canal de contato

O `get-contact-channel` exemplo a seguir lista os detalhes de um canal de contato.

```
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"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetContactChannel](#) na Referência de AWS CLI Comandos.

get-contact-policy

O código de exemplo a seguir mostra como usar get-contact-policy.

AWS CLI

Para listar as políticas de recursos de um contato

O get-contact-policy exemplo a seguir lista as políticas de recursos associadas ao contato especificado.

```
aws ssm-contacts get-contact-policy \  
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Saída:

```
{  
  "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",  
  "Policy": "{\n\"Version\": \"2012-10-17\", \"Statement\": [{\n\"Sid\":  
\n\"SharePolicyForDocumentationDralia\", \"Effect\": \"Allow\", \"Principal\":
```

```
{\"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/*\"]}]}
```

Para obter mais informações, consulte Como [trabalhar com contatos compartilhados e planos de resposta](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetContactPolicy](#) na Referência de AWS CLI Comandos.

get-contact

O código de exemplo a seguir mostra como usar get-contact.

AWS CLI

Exemplo 1: Para descrever um plano de contato

O get-contact exemplo a seguir descreve um contato.

```
aws ssm-contacts get-contact \
  --contact-id arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam
```

Saída:

```
{
  "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
      }
    }
  ]
}
]
}
}
}

```

Exemplo 2: Para descrever um plano de escalonamento

O `get-contact` exemplo a seguir descreve um plano de escalonamento.

```

aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"

```

Saída:

```
{
```

```
"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
            }
          }
        ]
      }
    ]
  }
]
```

```
}
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [GetContact](#) na Referência de AWS CLI Comandos.

list-contact-channels

O código de exemplo a seguir mostra como usar `list-contact-channels`.

AWS CLI

Para listar os canais de contato de um contato

O `list-contact-channels` exemplo a seguir lista os canais de contato disponíveis do contato especificado.

```
aws ssm-contacts list-contact-channels \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Saída:

```
{
  [
    {
      "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"
  }
]
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListContactChannels](#) na Referência de AWS CLI Comandos.

list-contacts

O código de exemplo a seguir mostra como usar `list-contacts`.

AWS CLI

Para listar todos os planos e contatos de escalonamento

O `list-contacts` exemplo a seguir lista os contatos e os planos de escalonamento em sua conta.

```
aws ssm-contacts list-contacts
```

Saída:

```
{
  "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/
alejrr",
      "Alias": "alejrr",
      "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"
}
]
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListContacts](#) na Referência de AWS CLI Comandos.

list-engagements

O código de exemplo a seguir mostra como usar `list-engagements`.

AWS CLI

Para listar todos os compromissos

O `list-engagements` exemplo a seguir lista os compromissos com planos e contatos de escalonamento. Você também pode listar compromissos para um único incidente.

```
aws ssm-contacts list-engagements
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListEngagements](#) na Referência de AWS CLI Comandos.

list-page-receipts

O código de exemplo a seguir mostra como usar `list-page-receipts`.

AWS CLI

Para listar os recibos da página

O `command-name` exemplo a seguir lista se uma página foi recebida ou não por um contato.

```
aws ssm-contacts list-page-receipts \  
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/  
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListPageReceipts](#) na Referência de AWS CLI Comandos.

list-pages-by-contact

O código de exemplo a seguir mostra como usar `list-pages-by-contact`.

AWS CLI

Para listar páginas por contato

O `list-pages-by-contact` exemplo a seguir lista todas as páginas do contato especificado.

```
aws ssm-contacts list-pages-by-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListPagesByContact](#) na Referência de AWS CLI Comandos.

list-pages-by-engagement

O código de exemplo a seguir mostra como usar `list-pages-by-engagement`.

AWS CLI

Listar páginas para canais de contato iniciados a partir de um engajamento.

O `list-pages-by-engagement` exemplo a seguir lista as páginas que ocorreram ao engajar o plano de engajamento definido.

```
aws ssm-contacts list-pages-by-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
akuam/78a29753-3674-4ac5-9f83-0468563567f0"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListPagesByEngagement](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um contato

O `list-tags-for-resource` exemplo a seguir lista as tags do contato especificado.

```
aws ssm-contacts list-tags-for-resource \
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Saída:

```
{
  "Tags": [
    {
      "Key": "group1",
      "Value": "1"
    }
  ]
}
```

```

    }
  ]
}

```

Para obter mais informações, consulte [Marcação](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-contact-policy

O código de exemplo a seguir mostra como usar `put-contact-policy`.

AWS CLI

Para compartilhar um contato e compromissos

O `put-contact-policy` exemplo a seguir adiciona uma política de recursos ao contato Akua que compartilha o contato e os compromissos relacionados com o diretor.

```

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/*\\\"]}]}"

```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com contatos compartilhados e planos de resposta](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [PutContactPolicy](#) na Referência de AWS CLI Comandos.

send-activation-code

O código de exemplo a seguir mostra como usar `send-activation-code`.

AWS CLI

Para enviar um código de ativação

O `send-activation-code` exemplo a seguir envia um código de ativação e uma mensagem para o canal de contato especificado.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [SendActivationCode](#) na Referência de AWS CLI Comandos.

start-engagement

O código de exemplo a seguir mostra como usar `start-engagement`.

AWS CLI

Exemplo 1: Para paginar os canais de contato de um contato

As `start-engagement` páginas a seguir entram em contato com os canais de contato. Remetente, assunto, assunto público e conteúdo público estão todos livres de campos. O Incident Manager envia o assunto e o conteúdo para os canais fornecidos VOICE ou de EMAIL contato. O Incident Manager envia o assunto público e o conteúdo público para os canais de contato fornecidos. SMS O remetente é usado para rastrear quem iniciou o engajamento.

```
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"
```

Saída:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

Exemplo 2: Incluir um contato no plano de escalonamento fornecido.

O seguinte `start-engagement` envolve os contatos por meio de um plano de escalonamento. Cada contato é paginado de acordo com seu plano de engajamento.

```
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"
```

Saída:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"  
}
```

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [StartEngagement](#) na Referência de AWS CLI Comandos.

stop-engagement

O código de exemplo a seguir mostra como usar `stop-engagement`.

AWS CLI

Para interromper um noivado

O `stop-engagement` exemplo a seguir impede que um engajamento pague outros contatos e canais de contato.

```
aws ssm-contacts stop-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```


Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [StopEngagement](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um contato

O `tag-resource` exemplo a seguir marca um contato especificado com o par de valores-chave de tag fornecido.

```
aws ssm-contacts tag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tags '[{"Key":"group1","Value":"1"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um contato

O `untag-resource` exemplo a seguir remove a tag `group1` do contato especificado.

```
aws ssm-contacts untag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tag-keys "group1"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-contact-channel

O código de exemplo a seguir mostra como usar `update-contact-channel`.

AWS CLI

Para atualizar um canal de contato

O `update-contact-channel` exemplo a seguir atualiza o nome e o endereço de entrega de um canal de contato.

```
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"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateContactChannel](#) na Referência de AWS CLI Comandos.

update-contact

O código de exemplo a seguir mostra como usar `update-contact`.

AWS CLI

Para atualizar o plano de engajamento do contato

O `update-contact` exemplo a seguir atualiza o plano de engajamento do contato Akua para incluir os três tipos de canais de contatos. Isso é feito depois de criar canais de contato para Akua.

```
aws ssm-contacts update-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact-  
channel/akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad" \  
  --engagement-plan "voice"
```

```
--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 }]}]}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Contatos](#) no Guia do usuário do Incident Manager.

- Para API obter detalhes, consulte [UpdateContact](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Inspector usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Inspector.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-attributes-to-findings

O código de exemplo a seguir mostra como usar `add-attributes-to-findings`.

AWS CLI

Para adicionar atributos às descobertas

O `add-attribute-to-finding` comando a seguir atribui um atributo com a chave de `Example` e o valor de `example` à descoberta com o ARN `dearn: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
```

Saída:

```
{
  "failedItems": {}
}
```

Para obter mais informações, consulte as descobertas do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [AddAttributesToFindings](#) na Referência de AWS CLI Comandos.

create-assessment-target

O código de exemplo a seguir mostra como usar `create-assessment-target`.

AWS CLI

Para criar uma meta de avaliação

O `create-assessment-target` comando a seguir cria um destino de avaliação chamado `ExampleAssessmentTarget` usando o grupo de recursos com o ARN `dearn: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
```

Saída:

```
{
```

```
"assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX"
}
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [CreateAssessmentTarget](#) na Referência de AWS CLI Comandos.

create-assessment-template

O código de exemplo a seguir mostra como usar `create-assessment-template`.

AWS CLI

Para criar um modelo de avaliação

O `create-assessment-template` comando a seguir cria um modelo de avaliação chamado `ExampleAssessmentTemplate` para o alvo de avaliação com o ARN `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
```

Saída:

```
{
  "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T"
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [CreateAssessmentTemplate](#) na Referência de AWS CLI Comandos.

create-resource-group

O código de exemplo a seguir mostra como usar `create-resource-group`.

AWS CLI

Para criar um grupo de recursos

O `create-resource-group` comando a seguir cria um grupo de recursos usando a chave de tag `Name` e o valor de exemplo:

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

Saída:

```
{
  "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv"
}
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [CreateResourceGroup](#) na Referência de AWS CLI Comandos.

delete-assessment-run

O código de exemplo a seguir mostra como usar `delete-assessment-run`.

AWS CLI

Para excluir uma execução de avaliação

O `delete-assessment-run` comando a seguir exclui a avaliação executada com o 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
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DeleteAssessmentRun](#) na Referência de AWS CLI Comandos.

delete-assessment-target

O código de exemplo a seguir mostra como usar delete-assessment-target.

AWS CLI

Para excluir um destino de avaliação

O delete-assessment-target comando a seguir exclui o destino da avaliação com o ARN `dearn: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
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DeleteAssessmentTarget](#) na Referência de AWS CLI Comandos.

delete-assessment-template

O código de exemplo a seguir mostra como usar delete-assessment-template.

AWS CLI

Para excluir um modelo de avaliação

O delete-assessment-template comando a seguir exclui o modelo de avaliação com o ARN `dearn: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
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DeleteAssessmentTemplate](#) na Referência de AWS CLI Comandos.

describe-assessment-runs

O código de exemplo a seguir mostra como usar `describe-assessment-runs`.

AWS CLI

Para descrever os ensaios de avaliação

O `describe-assessment-run` comando a seguir descreve uma avaliação executada com o 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
```

Saída:

```
{
  "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": {}  
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeAssessmentRuns](#) na Referência de AWS CLI Comandos.

describe-assessment-targets

O código de exemplo a seguir mostra como usar `describe-assessment-targets`.

AWS CLI

Para descrever as metas de avaliação

O `describe-assessment-targets` comando a seguir descreve o alvo da avaliação com o ARN `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
```

Saída:

```
{
  "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": {}
}
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeAssessmentTargets](#) na Referência de AWS CLI Comandos.

describe-assessment-templates

O código de exemplo a seguir mostra como usar `describe-assessment-templates`.

AWS CLI

Para descrever os modelos de avaliação

O `describe-assessment-templates` comando a seguir descreve o modelo de avaliação com o ARN `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw`:

```
aws inspector describe-assessment-templates --assessment-template-arns arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw
```

Saída:

```
{
  "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": {}
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeAssessmentTemplates](#) na Referência de AWS CLI Comandos.

describe-cross-account-access-role

O código de exemplo a seguir mostra como usar `describe-cross-account-access-role`.

AWS CLI

Para descrever a função de acesso entre contas

O `describe-cross-account-access-role` comando a seguir descreve a IAM função que permite ao Amazon Inspector acessar sua AWS conta:

```
aws inspector describe-cross-account-access-role
```

Saída:

```
{
  "registeredAt": 1458069182.826,
  "roleArn": "arn:aws:iam::123456789012:role/inspector",
  "valid": true
}
```

Para obter mais informações, consulte [Configurando o Amazon Inspector](#) no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeCrossAccountAccessRole](#) na Referência de AWS CLI Comandos.

describe-findings

O código de exemplo a seguir mostra como usar `describe-findings`.

AWS CLI

Para descrever as descobertas

O `describe-findings` comando a seguir descreve a descoberta com o 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
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte as descobertas do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeFindings](#) na Referência de AWS CLI Comandos.

describe-resource-groups

O código de exemplo a seguir mostra como usar `describe-resource-groups`.

AWS CLI

Para descrever grupos de recursos

O `describe-resource-groups` comando a seguir descreve o grupo de recursos com o ARN `dearn: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
```

Saída:

```
{
  "failedItems": {},
  "resourceGroups": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-
PyGXopAI",
      "createdAt": 1458074191.098,
      "tags": [
        {
          "key": "Name",
          "value": "example"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeResourceGroups](#) na Referência de AWS CLI Comandos.

describe-rules-packages

O código de exemplo a seguir mostra como usar `describe-rules-packages`.

AWS CLI

Para descrever pacotes de regras

O `describe-rules-packages` comando a seguir descreve o pacote de regras com o ARN `dearn: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
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Pacotes e regras de regras do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [DescribeRulesPackages](#) na Referência de AWS CLI Comandos.

get-telemetry-metadata

O código de exemplo a seguir mostra como usar `get-telemetry-metadata`.

AWS CLI

Para obter os metadados de telemetria

O `get-telemetry-metadata` comando a seguir gera informações sobre os dados coletados para a avaliação executada com o 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
```

Saída:

```
{
  "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"  
    }  
]  
}
```

- Para API obter detalhes, consulte [GetTelemetryMetadata](#) na Referência de AWS CLI Comandos.

list-assessment-run-agents

O código de exemplo a seguir mostra como usar `list-assessment-run-agents`.

AWS CLI

Para listar agentes de execução de avaliação

O `list-assessment-run-agents` comando a seguir lista os agentes da avaliação executada com o especificado 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
```

Saída:

```
{
  "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"
    }
  ]
}
]
```

Para obter mais informações, consulte [AWS Agentes](#) no Guia do Usuário do Amazon Inspector.

- Para API obter detalhes, consulte [ListAssessmentRunAgents](#) na Referência de AWS CLI Comandos.

list-assessment-runs

O código de exemplo a seguir mostra como usar `list-assessment-runs`.

AWS CLI

Para listar as execuções de avaliação

O `list-assessment-runs` comando a seguir lista todas as execuções de avaliação existentes.

```
aws inspector list-assessment-runs
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Modelos de avaliação e execuções de avaliação do Amazon Inspector no Guia](#) do usuário do Amazon Inspector.

- Para API obter detalhes, consulte [ListAssessmentRuns](#) na Referência de AWS CLI Comandos.

list-assessment-targets

O código de exemplo a seguir mostra como usar `list-assessment-targets`.

AWS CLI

Para listar os alvos da avaliação

O `list-assessment-targets` comando a seguir lista todas as metas de avaliação existentes:

```
aws inspector list-assessment-targets
```


Saída:

```
{
  "assessmentTargetArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
  ]
}
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [ListAssessmentTargets](#) na Referência de AWS CLI Comandos.

list-assessment-templates

O código de exemplo a seguir mostra como usar `list-assessment-templates`.

AWS CLI

Para listar modelos de avaliação

O `list-assessment-templates` comando a seguir lista todos os modelos de avaliação existentes:

```
aws inspector list-assessment-templates
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [ListAssessmentTemplates](#) na Referência de AWS CLI Comandos.

list-event-subscriptions

O código de exemplo a seguir mostra como usar `list-event-subscriptions`.

AWS CLI

Para listar assinaturas de eventos

O `list-event-subscriptions` comando a seguir lista todas as assinaturas de eventos para o modelo de avaliação com o ARN de: `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
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [ListEventSubscriptions](#) na Referência de AWS CLI Comandos.

list-findings

O código de exemplo a seguir mostra como usar `list-findings`.

AWS CLI

Para listar as descobertas

O `list-findings` comando a seguir lista todas as descobertas geradas:

```
aws inspector list-findings
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte as descobertas do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [ListFindings](#) na Referência de AWS CLI Comandos.

list-rules-packages

O código de exemplo a seguir mostra como usar `list-rules-packages`.

AWS CLI

Para listar pacotes de regras

O `list-rules-packages` comando a seguir lista todos os pacotes de regras do Inspector disponíveis:

```
aws inspector list-rules-packages
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte Pacotes e regras de regras do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [ListRulesPackages](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para o recurso

O `list-tags-for-resource` comando a seguir lista todas as tags associadas ao modelo de avaliação com o ARN `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
```

Saída:

```
{
  "tags": [
    {
      "key": "Name",
      "value": "Example"
    }
  ]
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

preview-agents

O código de exemplo a seguir mostra como usar `preview-agents`.

AWS CLI

Para pré-visualizar agentes

O `preview-agents` comando a seguir visualiza os agentes instalados nas EC2 instâncias que fazem parte da meta de avaliação com o ARN `dearn: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
```

Saída:

```
{
  "agentPreviews": [
    {
      "agentId": "i-49113b93"
    }
  ]
}
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [PreviewAgents](#) na Referência de AWS CLI Comandos.

register-cross-account-access-role

O código de exemplo a seguir mostra como usar `register-cross-account-access-role`.

AWS CLI

Para registrar a função de acesso entre contas

O `register-cross-account-access-role` comando a seguir registra a IAM função com a `arn:aws:iam::123456789012:role/inspector` que o ARN Amazon Inspector usa para listar EC2 suas instâncias no início da execução da avaliação ou quando você chama o comando `preview-agents`:

```
aws inspector register-cross-account-access-role --role-arn arn:aws:iam::123456789012:role/inspector
```

Para obter mais informações, consulte [Configurando o Amazon Inspector](#) no guia do Amazon Inspector.

- Para API obter detalhes, consulte [RegisterCrossAccountAccessRole](#) na Referência de AWS CLI Comandos.

remove-attributes-from-findings

O código de exemplo a seguir mostra como usar `remove-attributes-from-findings`.

AWS CLI

Para remover atributos das descobertas

O `remove-attributes-from-finding` comando a seguir remove o atributo com a chave de `Example` e o valor `example` de da descoberta com o ARN `dearn: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
```

Saída:

```
{
  "failedItems": {}
}
```

Para obter mais informações, consulte [as descobertas do Amazon Inspector](#) no guia do Amazon Inspector.

- Para API obter detalhes, consulte [RemoveAttributesFromFindings](#) na Referência de AWS CLI Comandos.

set-tags-for-resource

O código de exemplo a seguir mostra como usar `set-tags-for-resource`.

AWS CLI

Para definir tags para um recurso

O `set-tags-for-resource` comando a seguir define a tag com a chave de `Example` e o valor de `example` para o modelo de avaliação com o ARN `dearn: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
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [SetTagsForResource](#) na Referência de AWS CLI Comandos.

start-assessment-run

O código de exemplo a seguir mostra como usar `start-assessment-run`.

AWS CLI

Para iniciar uma execução de avaliação

O `start-assessment-run` comando a seguir inicia a execução da avaliação nomeada `examplerun` usando o modelo de avaliação com o ARN `dearn: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
```

Saída:

```
{
  "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY"
}
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [StartAssessmentRun](#) na Referência de AWS CLI Comandos.

stop-assessment-run

O código de exemplo a seguir mostra como usar `stop-assessment-run`.

AWS CLI

Para interromper uma execução de avaliação

O `stop-assessment-run` comando a seguir interrompe a execução da avaliação com o ARN `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY`:

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-
west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY
```

Para obter mais informações, consulte Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [StopAssessmentRun](#) na Referência de AWS CLI Comandos.

subscribe-to-event

O código de exemplo a seguir mostra como usar `subscribe-to-event`.

AWS CLI

Para se inscrever em um evento

O exemplo a seguir permite o processo de envio de SNS notificações da Amazon sobre o `ASSESSMENT_RUN_COMPLETED` evento para o tópico com o ARN de `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia](#) do Amazon Inspector.

- Para API obter detalhes, consulte [SubscribeToEvent](#) na Referência de AWS CLI Comandos.

unsubscribe-from-event

O código de exemplo a seguir mostra como usar unsubscribe-from-event.

AWS CLI

Para cancelar a inscrição em um evento

O unsubscribe-from-event comando a seguir desativa o processo de envio de SNS notificações da Amazon sobre o ASSESSMENT_RUN_COMPLETED evento para o tópico com o ARN de `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
```

Para obter mais informações, consulte [Modelos de avaliação e execuções de avaliação do Amazon Inspector no guia](#) do Amazon Inspector.

- Para API obter detalhes, consulte [UnsubscribeFromEvent](#) na Referência de AWS CLI Comandos.

update-assessment-target

O código de exemplo a seguir mostra como usar update-assessment-target.

AWS CLI

Para atualizar uma meta de avaliação

O `update-assessment-target` comando a seguir atualiza o destino da avaliação com o ARN de `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX` e o nome de `Example`, e o grupo de recursos com o ARN de `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
```

Para obter mais informações, consulte as metas de avaliação do Amazon Inspector no guia do Amazon Inspector.

- Para API obter detalhes, consulte [UpdateAssessmentTarget](#) na Referência de AWS CLI Comandos.

AWS IoT exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-certificate-transfer

O código de exemplo a seguir mostra como usar `accept-certificate-transfer`.

AWS CLI

Para aceitar um certificado de dispositivo transferido de uma AWS conta diferente

O `accept-certificate-transfer` exemplo a seguir aceita um certificado de dispositivo transferido de outra AWS conta. O certificado é identificado por seu ID.

```
aws iot accept-certificate-transfer \  
  --certificate-  
  id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Transferir um certificado para outra conta](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [AcceptCertificateTransfer](#) na Referência de AWS CLI Comandos.

add-thing-to-billing-group

O código de exemplo a seguir mostra como usar `add-thing-to-billing-group`.

AWS CLI

Exemplo 1: adicionar algo pelo nome a um grupo de cobrança

O `add-thing-to-billing-group` exemplo a seguir adiciona a coisa chamada `MyLightBulb` ao grupo de cobrança chamado `GroupOne`.

```
aws iot add-thing-to-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyLightBulb
```

Este comando não produz saída.

Exemplo 2: Para adicionar um item ARN a um grupo de cobrança

O `add-thing-to-billing-group` exemplo a seguir adiciona uma coisa com um especificado ARN a um grupo de cobrança com o especificado ARN. Especificar um ARN é útil se você trabalha com várias AWS regiões ou contas. Isso pode ajudar a garantir que você esteja adicionando à região e à conta corretas.

```
aws iot add-thing-to-thing-group \  
  --arn arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb
```

```
--billing-group-arn "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne" \  
--thing-arn "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb"
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [AddThingToBillingGroup](#) na Referência de AWS CLI Comandos.

add-thing-to-thing-group

O código de exemplo a seguir mostra como usar add-thing-to-thing-group.

AWS CLI

Para adicionar algo a um grupo

O add-thing-to-thing-group exemplo a seguir adiciona a coisa especificada ao grupo de coisas especificado.

```
aws iot add-thing-to-thing-group \  
--thing-name MyLightBulb \  
--thing-group-name LightBulbs
```

Este comando não produz saída.

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [AddThingToThingGroup](#) na Referência de AWS CLI Comandos.

associate-targets-with-job

O código de exemplo a seguir mostra como usar associate-targets-with-job.

AWS CLI

Associar um grupo de coisas a um trabalho contínuo

O `associate-targets-with-job` exemplo a seguir associa o grupo de coisas especificado ao trabalho contínuo especificado.

```
aws iot associate-targets-with-job \  
  --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --job-id "example-job-04"
```

Saída:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [AssociateTargetsWithJob](#) na Referência de AWS CLI Comandos.

attach-policy

O código de exemplo a seguir mostra como usar `attach-policy`.

AWS CLI

Exemplo 1: Para anexar uma política a um grupo de coisas

O `attach-policy` exemplo a seguir anexa a política especificada a um grupo de coisas identificado por itsARN.

```
aws iot attach-policy \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --policy-name "UpdateDeviceCertPolicy"
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

Exemplo 2: Para anexar uma política a um certificado

O `attach-policy` exemplo a seguir anexa a política `UpdateDeviceCertPolicy` ao principal especificado por um certificado.

```
aws iot attach-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --target "arn:aws:iot:us-west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Anexar uma política de AWS IoT a um certificado de dispositivo no Guia](#) para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [AttachPolicy](#) na Referência de AWS CLI Comandos.

attach-security-profile

O código de exemplo a seguir mostra como usar `attach-security-profile`.

AWS CLI

Para associar um perfil de segurança a todos os dispositivos não registrados

O `attach-security-profile` exemplo a seguir associa o perfil de segurança do AWS IoT Device Defender `Testprofile` nomeado a todos os dispositivos não registrados `us-west-2` na região para essa conta. AWS

```
aws iot attach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
```

Este comando não produz saída.

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [AttachSecurityProfile](#) na Referência de AWS CLI Comandos.

attach-thing-principal

O código de exemplo a seguir mostra como usar `attach-thing-principal`.

AWS CLI

Para anexar um certificado à sua coisa

O `attach-thing-principal` exemplo a seguir anexa um certificado à `MyTemperatureSensor` coisa. O certificado é identificado por um ARN. Você pode encontrar o ARN para um certificado no console de AWS IoT.

```
aws iot attach-thing-principal \  
  --thing-name MyTemperatureSensor \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

Este comando não produz saída.

Para acessar mais informações, consulte [Como gerenciar objetos com o registro](#) no Guia do desenvolvedor do AWS IoT.

- Para API obter detalhes, consulte [AttachThingPrincipal](#) na Referência de AWS CLI Comandos.

cancel-audit-mitigation-actions-task

O código de exemplo a seguir mostra como usar `cancel-audit-mitigation-actions-task`.

AWS CLI

Para cancelar uma tarefa de ações de mitigação de auditoria

O `cancel-audit-mitigation-actions-task` exemplo a seguir cancela a aplicação de ações de mitigação para a tarefa especificada. Você não pode cancelar tarefas que já foram concluídas.

```
aws iot cancel-audit-mitigation-actions-task  
  --task-id "myActionsTaskId"
```

Este comando não produz saída.

Para obter mais informações, consulte [CancelAuditMitigationActionsTask \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CancelAuditMitigationActionsTask](#) na Referência de AWS CLI Comandos.

cancel-audit-task

O código de exemplo a seguir mostra como usar `cancel-audit-task`.

AWS CLI

Para cancelar uma tarefa de auditoria

O `cancel-audit-task` exemplo a seguir cancela uma tarefa de auditoria com o ID de tarefa especificado. Você não pode cancelar uma tarefa que está concluída.

```
aws iot cancel-audit-task \  
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Este comando não produz saída.

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CancelAuditTask](#) na Referência de AWS CLI Comandos.

cancel-certificate-transfer

O código de exemplo a seguir mostra como usar `cancel-certificate-transfer`.

AWS CLI

Para cancelar a transferência de um certificado para uma AWS conta diferente

O `cancel-certificate-transfer` exemplo a seguir cancela a transferência da transferência de certificado especificada. O certificado é identificado por uma ID de certificado. Você pode encontrar a ID de um certificado no console de AWS IoT.

```
aws iot cancel-certificate-transfer \  
  --certificate-  
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

Este comando não produz saída.

Para obter mais informações, consulte [Transferir um certificado para outra conta](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [CancelCertificateTransfer](#) na Referência de AWS CLI Comandos.

cancel-job-execution

O código de exemplo a seguir mostra como usar `cancel-job-execution`.

AWS CLI

Para cancelar a execução de um trabalho em um dispositivo

O `cancel-job-execution` exemplo a seguir cancela a execução do trabalho especificado em um dispositivo. Se o trabalho não estiver no QUEUED estado, você deverá adicionar o `--force` parâmetro.

```
aws iot cancel-job-execution \  
  --job-id "example-job-03" \  
  --thing-name "MyRPi"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CancelJobExecution](#) na Referência de AWS CLI Comandos.

cancel-job

O código de exemplo a seguir mostra como usar `cancel-job`.

AWS CLI

Para cancelar um trabalho

O `cancel-job` exemplo a seguir cancela o trabalho especificado.

```
aws iot cancel-job \  
  --job-job "example-job-03"
```

Saída:

```
{
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",
  "jobId": "example-job-03",
  "description": "example job test"
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CancelJob](#) na Referência de AWS CLI Comandos.

clear-default-authorizer

O código de exemplo a seguir mostra como usar `clear-default-authorizer`.

AWS CLI

Para limpar o autorizador padrão

O `clear-default-authorizer` exemplo a seguir limpa o autorizador personalizado padrão atualmente configurado. Depois de executar esse comando, não há um autorizador padrão. Ao usar um autorizador personalizado, você deve especificá-lo pelo nome nos cabeçalhos da HTTP solicitação.

```
aws iot clear-default-authorizer
```

Este comando não produz saída.

Para obter mais informações, consulte [ClearDefaultAuthorizer](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ClearDefaultAuthorizer](#) na Referência de AWS CLI Comandos.

confirm-topic-rule-destination

O código de exemplo a seguir mostra como usar `confirm-topic-rule-destination`.

AWS CLI

Para confirmar o destino de uma regra de tópico

O `confirm-topic-rule-destination` exemplo a seguir confirma o destino de uma regra de tópico com um token de confirmação recebido em um HTTP endpoint.

```
aws iot confirm-topic-rule-destination \
  --confirmation-token "AYADeIcmtq-
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmXpYy1rZXkAREFxyY1E0Um1GeDg0V21BZWZ1VjZtZWFRVUJJUkt
aywpPqg8YEsa1LD4B40aJ2s1wEHKMybiF1Ro0ZzYisI0IvsLzQY5UmCkq3tV-3f7-
nKfosgIAAAAAADAAAEEAAAAAAAAAAAAAAAAAAAAAAi9RMgy-
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gT06n50NLJVpzyQR1UmPIj5sShqXEQGc0
iufgrzTePl8RZY0Wr006Aj9DiVzJZx-1iD6Pu-
G6PUw1ka07Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRMhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseR0LAN
PIetJ803Z4ILILF8xXlCdPGP-PV1d0XFemyL8g"
```

Este comando não produz saída.

Para obter mais informações, consulte [Confirmando o destino de uma regra de tópico](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ConfirmTopicRuleDestination](#) na Referência de AWS CLI Comandos.

create-audit-suppression

O código de exemplo a seguir mostra como usar `create-audit-suppression`.

AWS CLI

Para criar uma supressão da descoberta de auditoria

O `create-audit-suppression` exemplo a seguir cria uma supressão de constatação de auditoria para uma política chamada "virtualMachinePolicy" que foi sinalizada por ser excessivamente 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Auditoria encontrando supressões no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [CreateAuditSuppression](#) na Referência de AWS CLI Comandos.

create-authorizer

O código de exemplo a seguir mostra como usar create-authorizer.

AWS CLI

Para criar um autorizador personalizado

O create-authorizer exemplo a seguir cria um autorizador personalizado que usa a função Lambda especificada como parte de um serviço de autenticação personalizado.

```
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-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
Z+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTAgrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
zw0BKPeic0asNJpqT8PkBbRaKYLEJh5oo81NDHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx
50g2LW20y9R/dmqtDmJiVP97Z4GyKxPvwLYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
lQIDAQAB
-----END PUBLIC KEY-----"
```

Saída:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

Para obter mais informações, consulte [CreateAuthorizer](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [CreateAuthorizer](#) na Referência de AWS CLI Comandos.

create-billing-group

O código de exemplo a seguir mostra como usar `create-billing-group`.

AWS CLI

Para criar um grupo de cobrança

O `create-billing-group` exemplo a seguir cria um grupo de cobrança simples chamado `GroupOne`.

```
aws iot create-billing-group \  
  --billing-group-name GroupOne
```

Saída:

```
{  
  "billingGroupName": "GroupOne",  
  "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",  
  "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"  
}
```

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [CreateBillingGroup](#) na Referência de AWS CLI Comandos.

create-certificate-from-csr

O código de exemplo a seguir mostra como usar `create-certificate-from-csr`.

AWS CLI

Para criar um certificado de dispositivo a partir de uma solicitação de assinatura de certificado (CSR)

O `create-certificate-from-csr` exemplo a seguir cria um certificado de dispositivo a partir de um CSR. Você pode usar o `openssl` comando para criar um CSR.

```
aws iot create-certificate-from-csr \  
  --certificate-signing-request=file://certificate.csr
```

Saída:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/  
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificateId":  
  "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificatePem": "<certificate-text>"  
}
```

Para obter mais informações, consulte [CreateCertificateFromCSR](#) a Referência de AWS IoT. API

- Para API obter detalhes, consulte [CreateCertificateFromCsr](#) a Referência de AWS CLI Comandos.

create-custom-metric

O código de exemplo a seguir mostra como usar `create-custom-metric`.

AWS CLI

Para criar uma métrica personalizada publicada por seus dispositivos no Device Defender

O `create-custom-metric` exemplo a seguir cria uma métrica personalizada que mede a porcentagem da bateria.

```
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"
```

Saída:

```
{  
  "metricName": "batteryPercentage",
```

```
"metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage"
}
```

Para obter mais informações, consulte [Métricas personalizadas](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [CreateCustomMetric](#) na Referência de AWS CLI Comandos.

create-dimension

O código de exemplo a seguir mostra como usar `create-dimension`.

AWS CLI

Para criar uma dimensão

O seguinte `create-dimension` cria uma dimensão com um único filtro de tópico chamado `TopicFilterForAuthMessages`.

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

Saída:

```
{
  "name": "TopicFilterForAuthMessages",
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateDimension](#) na Referência de AWS CLI Comandos.

create-domain-configuration

O código de exemplo a seguir mostra como usar `create-domain-configuration`.

AWS CLI

Para criar uma configuração de domínio

O `create-domain-configuration` exemplo a seguir cria uma configuração AWS de domínio gerenciado com um tipo de serviço de DATA.

```
aws iot create-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --service-type "DATA"
```

Saída:

```
{  
  "domainConfigurationName": "additionalDataDomain",  
  "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"  
}
```

Para obter mais informações, consulte [Endpoints configuráveis no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateDomainConfiguration](#) na Referência de AWS CLI Comandos.

`create-dynamic-thing-group`

O código de exemplo a seguir mostra como usar `create-dynamic-thing-group`.

AWS CLI

Para criar um grupo dinâmico de coisas

O `create-dynamic-thing-group` exemplo a seguir cria um grupo dinâmico de coisas que contém qualquer coisa com um atributo de temperatura maior que 60 graus. Você deve habilitar a indexação da frota de AWS IoT antes de poder usar grupos dinâmicos de coisas.

```
aws iot create-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm" \  
  --query-string "attributes.temperature>60"
```


Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Dynamic Thing Groups](#) no Guia de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [CreateDynamicThingGroup](#) na Referência de AWS CLI Comandos.

create-job

O código de exemplo a seguir mostra como usar `create-job`.

AWS CLI

Exemplo 1: Para criar um trabalho

O `create-job` exemplo a seguir cria um trabalho simples de AWS IoT que envia um JSON documento para o 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
```

Saída:

```
{
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
  "jobId": "example-job-01",
  "description": "example job test"
}
```

```
}
```

Exemplo 2: Para criar um trabalho contínuo

O `create-job` exemplo a seguir cria um trabalho que continua sendo executado depois que as coisas especificadas como destinos concluírem o trabalho. Neste exemplo, o destino é um grupo de coisas, portanto, quando novos dispositivos são adicionados ao grupo, o trabalho contínuo é executado nessas coisas novas.

```
aws iot create-job --job-id "example-job-04" --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/" --document file: //example-job.json --description "exemplo de trabalho contínuo" --target-selection DeadBulbs CONTINUOUS
```

Saída:

```
{
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
  "jobId": "example-job-04",
  "description": "example continuous job"
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateJob](#) na Referência de AWS CLI Comandos.

create-keys-and-certificate

O código de exemplo a seguir mostra como usar `create-keys-and-certificate`.

AWS CLI

Para criar um RSA key pair e emitir um certificado X.509

O seguinte `create-keys-and-certificate` cria um par de chaves de 2048 bits e emite um certificado X.509 usando a RSA chave pública emitida. Como essa é a única vez que a AWS IoT fornece a chave privada para esse certificado, certifique-se de mantê-lo em um local seguro.

```
aws iot create-keys-and-certificate \
  --certificate-pem-outfile "myTest.cert.pem" \
```

```
--public-key-outfile "myTest.public.key" \
--private-key-outfile "myTest.private.key"
```

Saída:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificateId":
  "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMC
VVMxCzAJBgNVBAGEXAMPLEAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSEXAMPLE2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMaKGA1UEBhMCEXAMPLEJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAdBgkqhkiG9w0BCQEXAMPLE251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvYsWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZEXAMPLEL6G5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvaEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9qEXAMPEyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDEXAMPEBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEuuN/
dMAS3fyce8DW/4+EXAMPLEYjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqX0gHh/
xJTwo
+sGpWEXAMPLEDz18x0d2ka4tCzuWEXAMPLEEahJbYkCPUBSU8opVkr7qkEXAMPLE1DR6sx2Hocli00Ltu6Fkw91swQWEX
\GB3ZPrNh0PzQYvjUSTzEccyNCx2EXAMPLEvp9mQ0UXP6p1fgxwKRX2fEXAMPLEDa
\nhJLXkX3rHU2xbxJSq7D+XEXAMPLEcw+LyFhI5mgFR188eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----
END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security
reasons\n-----END RSA PRIVATE KEY-----\n"
  }
}
```

Para obter mais informações, consulte [Criar e registrar um certificado de dispositivo de IoT no Guia do desenvolvedor de AWS IoT.AWS](#)

- Para API obter detalhes, consulte [CreateKeysAndCertificate](#) na Referência de AWS CLI Comandos.

create-mitigation-action

O código de exemplo a seguir mostra como usar `create-mitigation-action`.

AWS CLI

Para criar uma ação de mitigação

O `create-mitigation-action` exemplo a seguir define uma ação de mitigação chamada `AddThingsToQuarantineGroup1Action` que, quando aplicada, move as coisas para o grupo de coisas chamado `QuarantineGroup1`. Essa ação substitui grupos dinâmicos de coisas.

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

Conteúdo de `params.json`:

```
{
  "actionName": "AddThingsToQuarantineGroup1Action",
  "actionParams": {
    "addThingsToThingGroupParams": {
      "thingGroupNames": [
        "QuarantineGroup1"
      ],
      "overrideDynamicGroups": true
    }
  },
  "roleArn": "arn:aws:iam::123456789012:role/service-role/MoveThingsToQuarantineGroupRole"
}
```

Saída:

```
{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/AddThingsToQuarantineGroup1Action",
  "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
}
```

Para obter mais informações, consulte [CreateMitigationAction \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateMitigationAction](#) na Referência de AWS CLI Comandos.

create-ota-update

O código de exemplo a seguir mostra como usar create-ota-update.

AWS CLI

Para criar uma OTA atualização para uso com o Amazon Free RTOS

O create-ota-update exemplo a seguir cria uma AWS IoT OTAUpdate em um grupo-alvo de coisas ou grupos. Isso faz parte de uma RTOS over-the-air atualização gratuita da Amazon que possibilita a implantação de novas imagens de firmware em um único dispositivo ou grupo de dispositivos.

```
aws iot create-ota-update \  
  --cli-input-json file://create-ota-update.json
```

Conteúdo de 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"
}

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [C reateOTAUpdate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [CreateOtaUpdate](#) na Referência de AWS CLI Comandos.

create-policy-version

O código de exemplo a seguir mostra como usar create-policy-version.

AWS CLI

Para atualizar uma política com uma nova versão

O create-policy-version exemplo a seguir atualiza uma definição de política, criando uma nova versão da política. Esse exemplo também torna a nova versão padrão.

```

aws iot create-policy-version \
  --policy-name UpdateDeviceCertPolicy \
  --policy-document file://policy.json \
  --set-as-default

```

Conteúdo de policy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:UpdateCertificate",
      "Resource": "*"
    }
  ]
}
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [CreatePolicyVersion](#) na Referência de AWS CLI Comandos.

create-policy

O código de exemplo a seguir mostra como usar create-policy.

AWS CLI

Para criar uma política de AWS IoT

O create-policy exemplo a seguir cria uma política AWS de IoT chamada.

TemperatureSensorPolicy O policy.json arquivo contém declarações que permitem ações de políticas de AWS IoT.

```
aws iot create-policy \
  --policy-name TemperatureSensorPolicy \
  --policy-document file://policy.json
```

Conteúdo de 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"
      ]
    }
  ]
}
```

Saída:

```
{
  "policyName": "TemperatureSensorPolicy",
```



```

    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
    "policyDocument": "{
      \"Version\": \"2012-10-17\",
      \"Statement\": [
        {
          \"Effect\": \"Allow\",
          \"Action\": [
            \"iot:Publish\",
            \"iot:Receive\"
          ],
          \"Resource\": [
            \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
            \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
          ]
        },
        {
          \"Effect\": \"Allow\",
          \"Action\": [
            \"iot:Subscribe\"
          ],
          \"Resource\": [
            \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_1\",
            \"arn:aws:iot:us-west-2:123456789012:topicfilter/topic_2\"
          ]
        },
        {
          \"Effect\": \"Allow\",
          \"Action\": [
            \"iot:Connect\"
          ],
          \"Resource\": [
            \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub\"
          ]
        }
      ]
    }",
    "policyVersionId": "1"
  }

```

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [CreatePolicy](#) na Referência de AWS CLI Comandos.


```
1bRyWU1r6qcpWBNBHjUoD7Hg0wD
\nnzMh4XRb2FQDsQFalkCSYmeL8IVC49sgPD90typ5uteGMTy62usAAUQdq/f
ZvrWg\n0kFpwMVnGKVKT7Kg0kKOLzKWOBB2Jm4/gmrJ\n-----END CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCg
KCAQEAwYSiPeJLSi6k8J4/msjq
\nUwCbfzer0iCQ2b5a2I5AtB08M2nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYN
pun8\n2pFpvf8KY8xPZ8ufsZDx1R+Fp8M+8iuZvEtGoC0/enEQUl1pqJzlnWNBilc54tA
\nGPoshrnYKxSpuxGn
v79fKF63/NirTgBjuMRtChNlimEXAMPLE3PcWYZVz/3ly4b9\nNPPRqdFlcPT24Sn68ZYiieaurC
+Z1VG1B6uoZU
D2MLAE3jrwKd7tGp0c4E8i9H40\n2D8ip0lWMC1utq/
lWAUhaKxDDgIZKty7D/648aRCpotluJT1UuqBWUSnA/h9
Ssop\nEQIDAQAB\n-----END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAwYSiPeJLSi6k8J4/
msjqUwCbfzer0iCQ2b5a2I5AtB08M2n
\nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYNpun82pFpvf8KY8xPZ8ufsZ
Dx1R+Fp8M+\n8iuZvEtGoC0/enEQUl1pqJzlnWNBilc54tAgPoshrnYKxSpuxGnv79fKF63/Nir
\nTgBjuMRtCh
NlimB7E9X8d3PcWYZVz/3ly4b9NPPRqdFlcPT24Sn68ZYiieaurC+Z
\n1VG1B6uoZUD2MLAE3jrwKd7tGp0c4E8i
9H402D8ip0lWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABAoIBAEAybn
QUtx9T2/nK\nTzT2pA4iugecxI4dz+DmT0XVxs5VJmrx/
nBSq6ejXExEpSIM04RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1kr0IPuxWebN0rRoPZr+1RSer+ww2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9Fd0UoQxdjHBHaM5PDMJ
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nnou0+91eTOP+/QfP7P8Zj0Ik02Xiv1RcVDyN/
E4QXPKuIkM/8vS8VK+
E9pATQ0MtB\n2lw8R/YU5AJd6jLEXAMPLEGU2UzRzInNWlLtkPPPqgqXXhxOf+mxByjcMalVJk0L
\nh0G2R0UCgY
EA+R0cHNHy/XbsP7Fih0hEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5S1MBp7S1idxak
bIn28fKfPn62DaemGCIOyDgLf+eUxBx
\nngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAXuWt\nnpzdWmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDxuxDe+V369Bianx8aZfyIsckA70wXW1w1h
\nngRC5rQ4X0gp3+Jmw7eA08LRYDjaN846+
Qbt02KcCgYAWS0UL51bijQR0ZwI0dz27\nnFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
```

```
BRtASW0Dab\naHXsDhVm2Jye+ESLoHMaCLoyCkT3118yqXIcEDStM07f01Ryag164EiJvSIrMfny\nNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2x0NZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXgh0eoTpC8VXkzcnzouTiQ
\nFQQSGfnp7Ioe6UIz23715pKduzSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nPRTIZQKBgA8MASXtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJcKAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nvdlc5cFQewWb4/
zqAtPdinkPLN94ileI
79XQdc7RlJ0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
    },
    "expiration": 1595955066.0
  }
}
```

Para obter mais informações, consulte [Provisionamento por usuário confiável no Guia](#) para desenvolvedores do AWS IoT Core.

- Para API obter detalhes, consulte [CreateProvisioningClaim](#) na Referência de AWS CLI Comandos.

create-provisioning-template-version

O código de exemplo a seguir mostra como usar `create-provisioning-template-version`.

AWS CLI

Para criar uma versão do modelo de aprovisionamento

O exemplo a seguir cria uma versão para o modelo de aprovisionamento especificado. O corpo da nova versão é fornecido no arquivo `template.json`.

```
aws iot create-provisioning-template-version \
  --template-name widget-template \
  --template-body file://template.json
```

Conteúdo de `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"]}
    }
  }
}

```

Saída:

```

{
  "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
  "templateName": "widget-template",
  "versionId": 2,
  "isDefaultVersion": false
}

```

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor](#) do IoT Core AWS .

- Para API obter detalhes, consulte [CreateProvisioningTemplateVersion](#) na Referência de AWS CLI Comandos.

create-provisioning-template

O código de exemplo a seguir mostra como usar create-provisioning-template.

AWS CLI

Para criar um modelo de provisionamento

O create-provisioning-template exemplo a seguir cria um modelo de provisionamento conforme definido pelo arquivo. `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
```

Conteúdo de `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"]}
    }
  }
}

```

Saída:

```

{
  "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
  "templateName": "widget-template",
  "defaultVersionId": 1
}

```

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor do IoT Core AWS](#).

- Para API obter detalhes, consulte [CreateProvisioningTemplate](#) na Referência de AWS CLI Comandos.

create-role-alias

O código de exemplo a seguir mostra como usar `create-role-alias`.

AWS CLI

Para criar um alias de função

O `create-role-alias` exemplo a seguir cria um alias de função chamado `LightBulbRole` para a função especificada.

```
aws iot create-role-alias \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Saída:

```
{  
  "roleAlias": "LightBulbRole",  
  "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"  
}
```

Para obter mais informações, consulte [CreateRoleAlias](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [CreateRoleAlias](#) na Referência de AWS CLI Comandos.

create-scheduled-audit

O código de exemplo a seguir mostra como usar `create-scheduled-audit`.

AWS CLI

Para criar uma auditoria agendada

O `create-scheduled-audit` exemplo a seguir cria uma auditoria agendada que é executada semanalmente, na quarta-feira, para verificar se os certificados da CA ou os certificados do dispositivo estão expirando.

```
aws iot create-scheduled-audit \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

```

--scheduled-audit-name WednesdayCertCheck \
--frequency WEEKLY \
--day-of-week WED \
--target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK

```

Saída:

```

{
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}

```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateScheduledAudit](#) na Referência de AWS CLI Comandos.

create-security-profile

O código de exemplo a seguir mostra como usar create-security-profile.

AWS CLI

Para criar um perfil de segurança

O create-security-profile exemplo a seguir cria um perfil de segurança que verifica se a largura de banda celular excede um limite ou se mais de 10 falhas de autorização ocorrem em um período de cinco minutos.

```

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}]]"
```

Saída:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue"
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateSecurityProfile](#) na Referência de AWS CLI Comandos.

create-stream

O código de exemplo a seguir mostra como usar `create-stream`.

AWS CLI

Para criar um stream para entregar um ou mais arquivos grandes em partes acima MQTT

O `create-stream` exemplo a seguir cria um fluxo para entregar um ou mais arquivos grandes em partes. MQTT Um stream transporta bytes de dados em partes ou blocos empacotados como MQTT mensagens de uma fonte como o S3. Você pode ter um ou mais arquivos associados a um fluxo.

```
aws iot create-stream \
  --cli-input-json file://create-stream.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [CreateStream](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [CreateStream](#) na Referência de AWS CLI Comandos.

create-thing-group

O código de exemplo a seguir mostra como usar `create-thing-group`.

AWS CLI

Exemplo 1: Para criar um grupo de coisas

O `create-thing-group` exemplo a seguir cria um grupo de coisas chamado `LightBulbs` com uma descrição e dois atributos.

```
aws iot create-thing-group \  
  --thing-group-name LightBulbs \  
  --thing-group-properties "thingGroupDescription=\"Generic bulb group\",  
  attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

Saída:

```
{  
  "thingGroupName": "LightBulbs",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",  
  "thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"  
}
```

Exemplo 2: Para criar um grupo de coisas que faça parte de um grupo principal

O seguinte `create-thing-group` cria um grupo de coisas chamado `HalogenBulbs` que tem um grupo de coisas pai chamado `LightBulbs`.

```
aws iot create-thing-group \  
  --thing-group-name HalogenBulbs \  
  --parent-group-name LightBulbs
```

Saída:

```
{  
  "thingGroupName": "HalogenBulbs",  
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",  
  "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"  
}
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [CreateThingGroup](#) na Referência de AWS CLI Comandos.

create-thing-type

O código de exemplo a seguir mostra como usar `create-thing-type`.

AWS CLI

Para definir um tipo de coisa

O `create-thing-type` exemplo a seguir define um tipo de coisa e atributos associados.

```
aws iot create-thing-type \  
  --thing-type-name "LightBulb" \  
  --thing-type-properties "thingTypeDescription=light bulb type,  
  searchableAttributes=wattage,model"
```

Saída:

```
{  
  "thingTypeName": "LightBulb",  
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",  
  "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"  
}
```

Para obter mais informações, consulte [Tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [CreateThingType](#) na Referência de AWS CLI Comandos.

create-thing

O código de exemplo a seguir mostra como usar `create-thing`.

AWS CLI

Exemplo 1: Para criar um registro de coisa no registro

O `create-thing` exemplo a seguir cria uma entrada para um dispositivo no registro de coisas da AWS IoT.

```
aws iot create-thing \  
  --thing-name SampleIoTThing
```

Saída:

```
{  
  "thingName": "SampleIoTThing",  
  "thingArn": "arn:aws:iot:us-west-2: 123456789012:thing/SampleIoTThing",  
  "thingId": " EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "  
}
```

Exemplo 2: Para definir uma coisa associada a um tipo de coisa

O `create-thing` exemplo a seguir cria uma coisa que tem o tipo de coisa especificado e seus atributos.

```
aws iot create-thing \  
  --thing-name "MyLightBulb" \  
  --thing-type-name "LightBulb" \  
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

Saída:

```
{  
  "thingName": "MyLightBulb",
```

```
"thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
"thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

Para obter mais informações, consulte [Como gerenciar coisas com o registro e os tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [CreateThing](#) na Referência de AWS CLI Comandos.

create-topic-rule-destination

O código de exemplo a seguir mostra como usar `create-topic-rule-destination`.

AWS CLI

Para criar um destino de regra de tópico

O `create-topic-rule-destination` exemplo a seguir cria um destino de regra de tópico para um HTTP endpoint.

```
aws iot create-topic-rule-destination \
  --destination-configuration httpUrlConfiguration={confirmationUrl=https://
  example.com}
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Criação de um destino de regra de tópico](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateTopicRuleDestination](#) na Referência de AWS CLI Comandos.

create-topic-rule

O código de exemplo a seguir mostra como usar `create-topic-rule`.

AWS CLI

Para criar uma regra que envie um SNS alerta da Amazon

O `create-topic-rule` exemplo a seguir cria uma regra que envia uma SNS mensagem à Amazon quando as leituras do nível de umidade do solo, encontradas na sombra de um dispositivo, são baixas.

```
aws iot create-topic-rule \  
  --rule-name "LowMoistureRule" \  
  --topic-rule-payload file://plant-rule.json
```

O exemplo exige que o JSON código a seguir seja salvo em um arquivo chamado `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"  
    }  
  ]  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma regra de AWS IoT no Guia para desenvolvedores de AWS IoT](#).

- Para API obter detalhes, consulte [CreateTopicRule](#) na Referência de AWS CLI Comandos.

delete-account-audit-configuration

O código de exemplo a seguir mostra como usar `delete-account-audit-configuration`.

AWS CLI

Para desativar todas as verificações de auditoria da sua AWS conta

O `delete-account-audit-configuration` exemplo a seguir restaura as configurações padrão do AWS IoT Device Defender para essa conta, desativando todas as verificações de auditoria e limpando os dados de configuração. Ele também exclui todas as auditorias agendadas para essa conta. Use esse comando com cuidado.

```
aws iot delete-account-audit-configuration \  
  --delete-scheduled-audits
```

Este comando não produz saída.

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteAccountAuditConfiguration](#) na Referência de AWS CLI Comandos.

delete-audit-suppression

O código de exemplo a seguir mostra como usar `delete-audit-suppression`.

AWS CLI

Para excluir uma supressão de descoberta de auditoria

O `delete-audit-suppression` exemplo a seguir exclui uma supressão de constatação de auditoria para `DEVICE __ CERTIFICATE _EXPIRING. CHECK`

```
aws iot delete-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --
```

```
--resource-identifier deviceCertificateId="c7691e<shortened>"
```

Este comando não produz saída.

Para obter mais informações, consulte [Auditoria encontrando supressões no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DeleteAuditSuppression](#) na Referência de AWS CLI Comandos.

delete-authorizer

O código de exemplo a seguir mostra como usar `delete-authorizer`.

AWS CLI

Para excluir um autorizador personalizado

O `delete-authorizer` exemplo a seguir exclui o autorizador chamado `CustomAuthorizer`. Um autorizador personalizado deve estar no INACTIVE estado antes que você possa excluí-lo.

```
aws iot delete-authorizer \  
  --authorizer-name CustomAuthorizer
```

Este comando não produz saída.

Para obter mais informações, consulte o [DeleteAuthorizer](#) Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteAuthorizer](#) na Referência de AWS CLI Comandos.

delete-billing-group

O código de exemplo a seguir mostra como usar `delete-billing-group`.

AWS CLI

Para excluir um grupo de cobrança

O `delete-billing-group` exemplo a seguir exclui o grupo de cobrança especificado. Você pode excluir um grupo de cobrança mesmo que ele contenha uma ou mais coisas.

```
aws iot delete-billing-group \  
  --billing-group-name MyBillingGroup
```

```
--billing-group-name BillingGroupTwo
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DeleteBillingGroup](#) na Referência de AWS CLI Comandos.

delete-ca-certificate

O código de exemplo a seguir mostra como usar `delete-ca-certificate`.

AWS CLI

Para excluir um certificado CA

O `delete-ca-certificate` exemplo a seguir exclui o certificado CA com o ID do certificado especificado.

```
aws iot delete-ca-certificate \  
  --certificate-  
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteCACertificate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DeleteCaCertificate](#) na Referência de AWS CLI Comandos.

delete-certificate

O código de exemplo a seguir mostra como usar `delete-certificate`.

AWS CLI

Para excluir um certificado de dispositivo

O `delete-certificate` exemplo a seguir exclui o certificado do dispositivo com a ID especificada.

```
aws iot delete-certificate \  
  --certificate-id
```

```
--certificate-  
id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteCertificate](#) a Referência de AWS IoT. API

- Para API obter detalhes, consulte [DeleteCertificate](#) na Referência de AWS CLI Comandos.

delete-custom-metric

O código de exemplo a seguir mostra como usar `delete-custom-metric`.

AWS CLI

Para excluir uma métrica personalizada

O `delete-custom-metric` exemplo a seguir exclui uma métrica personalizada.

```
aws iot delete-custom-metric \  
  --metric-name batteryPercentage \  
  --region us-east-1
```

Saída:

```
HTTP 200
```

Para obter mais informações, consulte [Métricas personalizadas](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [DeleteCustomMetric](#) na Referência de AWS CLI Comandos.

delete-dimension

O código de exemplo a seguir mostra como usar `delete-dimension`.

AWS CLI

Para excluir uma dimensão

O `delete-dimension` exemplo a seguir exclui uma dimensão chamada `TopicFilterForAuthMessages`.

```
aws iot delete-dimension \  
  --name TopicFilterForAuthMessages
```

Este comando não produz saída.

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteDimension](#) na Referência de AWS CLI Comandos.

delete-domain-configuration

O código de exemplo a seguir mostra como usar delete-domain-configuration.

AWS CLI

Para excluir uma configuração de domínio

O delete-domain-configuration exemplo a seguir exclui uma configuração de domínio chamada additionalDataDomain da sua AWS conta.

```
aws iot delete-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --domain-configuration-status "OK"
```

Este comando não produz saída.

Para obter mais informações, consulte [Endpoints configuráveis no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteDomainConfiguration](#) na Referência de AWS CLI Comandos.

delete-dynamic-thing-group

O código de exemplo a seguir mostra como usar delete-dynamic-thing-group.

AWS CLI

Para excluir um grupo dinâmico de coisas

O `delete-dynamic-thing-group` exemplo a seguir exclui o grupo dinâmico de coisas especificado.

```
aws iot delete-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm"
```

Este comando não produz saída.

Para obter mais informações, consulte [Dynamic Thing Groups](#) no Guia de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DeleteDynamicThingGroup](#) na Referência de AWS CLI Comandos.

delete-job-execution

O código de exemplo a seguir mostra como usar `delete-job-execution`.

AWS CLI

Para excluir uma execução de trabalho

O `delete-job-execution` exemplo a seguir exclui a execução do trabalho especificado em um dispositivo. Use `describe-job-execution` para obter o número da execução.

```
aws iot delete-job-execution \  
  --job-id "example-job-02" \  
  --thing-name "MyRaspberryPi" \  
  --execution-number 1
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteJobExecution](#) na Referência de AWS CLI Comandos.

delete-job

O código de exemplo a seguir mostra como usar `delete-job`.

AWS CLI

Para excluir um trabalho

O `delete-job` exemplo a seguir exclui o trabalho especificado. Ao especificar a `--force` opção, o trabalho é excluído mesmo que o status seja `IN_PROGRESS`.

```
aws iot delete-job \  
  --job-id "example-job-04" \  
  --force
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteJob](#) na Referência de AWS CLI Comandos.

`delete-mitigation-action`

O código de exemplo a seguir mostra como usar `delete-mitigation-action`.

AWS CLI

Para excluir uma ação de mitigação

O `delete-mitigation-action` exemplo a seguir exclui a ação de mitigação especificada.

```
aws iot delete-mitigation-action \  
  --action-name AddThingsToQuarantineGroup1Action
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteMitigationAction \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteMitigationAction](#) na Referência de AWS CLI Comandos.

`delete-ota-update`

O código de exemplo a seguir mostra como usar `delete-ota-update`.

AWS CLI

Para excluir uma OTA atualização

O `delete-ota-update` exemplo a seguir exclui a OTA atualização especificada.

```
aws iot delete-ota-update \  
  --ota-update-id ota12345 \  
  --delete-stream \  
  --force-delete-aws-job
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteOTAUpdate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DeleteOtaUpdate](#) na Referência de AWS CLI Comandos.

`delete-policy-version`

O código de exemplo a seguir mostra como usar `delete-policy-version`.

AWS CLI

Para excluir uma versão da política

O `delete-policy-version` exemplo a seguir exclui a versão 2 da política especificada da sua AWS conta.

```
aws iot delete-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas de AWS IoT no Guia do](#) desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeletePolicyVersion](#) na Referência de AWS CLI Comandos.

`delete-policy`

O código de exemplo a seguir mostra como usar `delete-policy`.

AWS CLI

Como excluir uma política

O `delete-policy` exemplo a seguir exclui a política especificada da sua AWS conta.

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

Este comando não produz saída.

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

`delete-provisioning-template-version`

O código de exemplo a seguir mostra como usar `delete-provisioning-template-version`.

AWS CLI

Para excluir uma versão do modelo de aprovisionamento

O `delete-provisioning-template-version` exemplo a seguir exclui a versão 2 do modelo de aprovisionamento especificado.

```
aws iot delete-provisioning-template-version \  
  --version-id 2 \  
  --template-name "widget-template"
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor](#) do IoT Core AWS .

- Para API obter detalhes, consulte [DeleteProvisioningTemplateVersion](#) na Referência de AWS CLI Comandos.

`delete-provisioning-template`

O código de exemplo a seguir mostra como usar `delete-provisioning-template`.

AWS CLI

Para excluir um modelo de provisionamento

O `delete-provisioning-template` exemplo a seguir exclui o modelo de provisionamento especificado.

```
aws iot delete-provisioning-template \  
  --template-name widget-template
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor](#) do IoT Core AWS .

- Para API obter detalhes, consulte [DeleteProvisioningTemplate](#) na Referência de AWS CLI Comandos.

delete-registration-code

O código de exemplo a seguir mostra como usar `delete-registration-code`.

AWS CLI

Para excluir seu código de registro

O `delete-registration-code` exemplo a seguir exclui um código de registro específico da conta de AWS IoT.

```
aws iot delete-registration-code
```

Este comando não produz saída.

Para obter mais informações, consulte [Use seu próprio certificado](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteRegistrationCode](#) na Referência de AWS CLI Comandos.

delete-role-alias

O código de exemplo a seguir mostra como usar `delete-role-alias`.

AWS CLI

Para excluir um alias de AWS função de IoT

O `delete-role-alias` exemplo a seguir exclui um alias de função de AWS IoT chamado `LightBulbRole`

```
aws iot delete-role-alias \  
  --role-alias LightBulbRole
```

Este comando não produz saída.

Para obter mais informações, consulte [Autorização de chamadas diretas para AWS serviços](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteRoleAlias](#) na Referência de AWS CLI Comandos.

`delete-scheduled-audit`

O código de exemplo a seguir mostra como usar `delete-scheduled-audit`.

AWS CLI

Para excluir uma auditoria agendada

O `delete-scheduled-audit` exemplo a seguir exclui a auditoria agendada do AWS IoT Device Defender chamada `AWSIoTDeviceDefenderDailyAudit`

```
aws iot delete-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Este comando não produz saída.

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteScheduledAudit](#) na Referência de AWS CLI Comandos.

`delete-security-profile`

O código de exemplo a seguir mostra como usar `delete-security-profile`.

AWS CLI

Para excluir um perfil de segurança

O `delete-security-profile` exemplo a seguir exclui um perfil de segurança chamado `PossibleIssue`.

```
aws iot delete-security-profile \  
  --security-profile-name PossibleIssue
```

Este comando não produz saída.

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteSecurityProfile](#) na Referência de AWS CLI Comandos.

`delete-stream`

O código de exemplo a seguir mostra como usar `delete-stream`.

AWS CLI

Para excluir um stream

O `delete-stream` exemplo a seguir exclui o fluxo especificado.

```
aws iot delete-stream \  
  --stream-id stream12345
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteStream](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DeleteStream](#) na Referência de AWS CLI Comandos.

`delete-thing-group`

O código de exemplo a seguir mostra como usar `delete-thing-group`.

AWS CLI

Para excluir um grupo de coisas

O `delete-thing-group` exemplo a seguir exclui o grupo de coisas especificado. Você não pode excluir um grupo de coisas se ele contiver grupos de coisas secundárias.

```
aws iot delete-thing-group \  
  --thing-group-name DefectiveBulbs
```

Este comando não produz saída.

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [DeleteThingGroup](#) na Referência de AWS CLI Comandos.

delete-thing-type

O código de exemplo a seguir mostra como usar `delete-thing-type`.

AWS CLI

Exemplo 1: Para excluir um tipo de coisa

O `delete-thing-type` exemplo a seguir exclui um tipo de item obsoleto.

```
aws iot delete-thing-type \  
  --thing-type-name "obsoleteThingType"
```

Este comando não produz saída.

Para obter mais informações, consulte [Tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [DeleteThingType](#) na Referência de AWS CLI Comandos.

delete-thing

O código de exemplo a seguir mostra como usar `delete-thing`.

AWS CLI

Para exibir informações detalhadas sobre uma coisa

O `delete-thing` exemplo a seguir exclui algo do registro de AWS IoT da sua conta. AWS

era `delete-thing --thing-name` FourthBulb

Este comando não produz saída.

Para acessar mais informações, consulte [Como gerenciar objetos com o registro](#) no Guia do desenvolvedor do AWS IoT.

- Para API obter detalhes, consulte [DeleteThing](#) na Referência de AWS CLI Comandos.

delete-topic-rule-destination

O código de exemplo a seguir mostra como usar `delete-topic-rule-destination`.

AWS CLI

Para excluir um destino de regra de tópico

O `delete-topic-rule-destination` exemplo a seguir exclui o destino da regra de tópico especificada.

```
aws iot delete-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um destino de regra de tópico](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteTopicRuleDestination](#) na Referência de AWS CLI Comandos.

delete-topic-rule

O código de exemplo a seguir mostra como usar `delete-topic-rule`.

AWS CLI

Para excluir uma regra

O `delete-topic-rule` exemplo a seguir exclui a regra especificada.

```
aws iot delete-topic-rule \  
  --rule-name
```

```
--rule-name "LowMoistureRule"
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo uma regra no Guia](#) para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DeleteTopicRule](#) na Referência de AWS CLI Comandos.

delete-v2-logging-level

O código de exemplo a seguir mostra como usar delete-v2-logging-level.

AWS CLI

Para excluir o nível de registro de um grupo de coisas

O delete-v2-logging-level exemplo a seguir exclui o nível de registro do grupo de coisas especificado.

```
aws iot delete-v2-logging-level \  
  --target-type THING_GROUP \  
  --target-name LightBulbs
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteV2 LoggingLevel](#) na AWS CLI Referência de Comandos.

deprecate-thing-type

O código de exemplo a seguir mostra como usar deprecate-thing-type.

AWS CLI

Exemplo 1: Para descontinuar um tipo de coisa

O deprecate-thing-type exemplo a seguir desaprova um tipo de coisa para que os usuários não possam associar nenhuma coisa nova a ela.

```
aws iot deprecate-thing-type \  
  --target-type THING_TYPE \  
  --target-name LightBulbs
```

```
--thing-type-name obsoleteThingType
```

Este comando não produz saída.

Exemplo 2: Para reverter a depreciação de um tipo de coisa

O `deprecate-thing-type` exemplo a seguir reverte a suspensão de uso de um tipo de item, o que possibilita que os usuários associem coisas novas a ele novamente.

```
aws iot deprecate-thing-type \  
  --thing-type-name obsoleteThingType \  
  --undo-deprecate
```

Este comando não produz saída.

Para obter mais informações, consulte [Tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [DeprecateThingType](#) na Referência de AWS CLI Comandos.

describe-account-audit-configuration

O código de exemplo a seguir mostra como usar `describe-account-audit-configuration`.

AWS CLI

Para visualizar as configurações de auditoria atuais

O `describe-account-audit-configuration` exemplo a seguir lista as configurações atuais da sua configuração de auditoria AWS do IoT Device Defender.

```
aws iot describe-account-audit-configuration
```

Saída:

```
{  
  "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
    }
}
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeAccountAuditConfiguration](#) na Referência de AWS CLI Comandos.

describe-audit-finding

O código de exemplo a seguir mostra como usar `describe-audit-finding`.

AWS CLI

Para listar detalhes de uma descoberta de auditoria

O `describe-audit-finding` exemplo a seguir lista os detalhes da descoberta de auditoria especificada do AWS IoT Device Defender. Uma auditoria pode produzir várias descobertas. Use o `list-audit-findings` comando para obter uma lista das descobertas de uma auditoria para obter `findingId` o.

```
aws iot describe-audit-finding \  
  --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Verificar resultados de auditoria \(comandos de auditoria\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeAuditFinding](#) na Referência de AWS CLI Comandos.

describe-audit-mitigation-actions-task

O código de exemplo a seguir mostra como usar `describe-audit-mitigation-actions-task`.

AWS CLI

Para mostrar os detalhes de uma tarefa de ações de mitigação de auditoria

O `describe-audit-mitigation-actions-task` exemplo a seguir mostra os detalhes da tarefa especificada, na qual o `ResetPolicyVersionAction` foi aplicado a uma descoberta. Os resultados incluem quando a tarefa começou e terminou, quantas descobertas foram direcionadas (e o resultado) e a definição da ação que é aplicada como parte dessa tarefa.

```
aws iot describe-audit-mitigation-actions-task \  
  --task-id ResetPolicyTask01
```

Saída:

```
{  
  "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"
          }
        }
      }
    ]
  }
}

```

Para obter mais informações, consulte [DescribeAuditMitigationActionsTask \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeAuditMitigationActionsTask](#) na Referência de AWS CLI Comandos.

describe-audit-suppression

O código de exemplo a seguir mostra como usar describe-audit-suppression.

AWS CLI

Para obter detalhes sobre a supressão de uma descoberta de auditoria

O describe-audit-suppression exemplo a seguir lista detalhes sobre a supressão de uma descoberta de auditoria.

```

aws iot describe-audit-task \
  --task-id "787ed873b69cb4d6cdbae6ddd06996c5"

```

Saída:

```
{
```

```
"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
  }
}
}
```

Para obter mais informações, consulte [Auditoria encontrando supressões no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DescribeAuditSuppression](#) na Referência de AWS CLI Comandos.

describe-audit-task

O código de exemplo a seguir mostra como usar `describe-audit-task`.

AWS CLI

Para obter informações sobre uma instância de auditoria

O `describe-audit-task` exemplo a seguir obtém informações sobre uma instância de uma auditoria do AWS IoT Device Defender. Se a auditoria for concluída, as estatísticas resumidas da execução serão incluídas nos resultados.

```
aws iot describe-audit-task \
```

```
--task-id a3aea009955e501a31b764abe1bebd3d
```

Saída:

```
{
  "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
    }
  }
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeAuditTask](#) na Referência de AWS CLI Comandos.

describe-authorizer

O código de exemplo a seguir mostra como usar `describe-authorizer`.

AWS CLI

Para obter informações sobre um autorizador personalizado

O `describe-authorizer` exemplo a seguir exibe detalhes do autorizador personalizado especificado.

```
aws iot describe-authorizer \
  --authorizer-name CustomAuthorizer
```

Saída:

```
{
  "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-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEA1uJ0B4lQPgG/1M6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAGTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzw0BKPeic0asNJpqT8PkBbRaKYLEJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfP0HDJQgID0XnZwAlNnZc0hCwIx\n50g2LW20y9R/
dmqtDmJiVP97Z4GykxPvwlyHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
KEY-----"
    },
    "status": "ACTIVE",
    "creationDate": 1571245658.069,
    "lastModifiedDate": 1571245658.069
  }
}
```

Para obter mais informações, consulte [DescribeAuthorizer](#) a Referência de AWS IoT. API

- Para API obter detalhes, consulte [DescribeAuthorizer](#) na Referência de AWS CLI Comandos.

describe-billing-group

O código de exemplo a seguir mostra como usar `describe-billing-group`.

AWS CLI

Para obter informações sobre um grupo de cobrança

O `describe-billing-group` exemplo a seguir obtém informações para o grupo de cobrança especificado.

```
aws iot describe-billing-group --billing-group-name GroupOne
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DescribeBillingGroup](#) na Referência de AWS CLI Comandos.

describe-ca-certificate

O código de exemplo a seguir mostra como usar `describe-ca-certificate`.

AWS CLI

Para obter detalhes sobre um certificado CA

O `describe-ca-certificate` exemplo a seguir exibe os detalhes do certificado CA especificado.

```
aws iot describe-ca-certificate \
```


--certificate-**id** **f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467**

Saída:

```
{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "status": "INACTIVE",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPWX18taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTA1VMTQ8wDQYDVQQKDAZBbWw6b24wHhcNMm0TI0MjEzMTU1WhcNMjkwOTIx
\nMjEzMTU1WjAeMQswCQYDVQQGEwJVUzEPMA0GA1UECgwGQW1hem9uMIIBIjANBgkq
\nhkIG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzd3R3ioalCS0MhFwFBrVGR036EK07UAF
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18Vz1FtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMaw0jhF4Lscu1KG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWWvUskzxn01F2WCshbEjvqY8fIwtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJw0QxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+n04jd5cqaoEPtzoEUEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJs1xwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3IJg0nd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcB1CFwH1yakwjJL1dR4yarnrfxgI
\nEqP4A0YVimAVoQ5FBwnloHe16+3qtDiblU9DeXBUctS55EcfREXAMPLEYtXdqU5C
\nU9ia4KAjV0dxW1+EFYmWx5eGeb0gDTNHBylV6B/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
    }
  }
}
```

Para obter mais informações, consulte [D escreveCACertificate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DescribeCaCertificate](#) na Referência de AWS CLI Comandos.

describe-certificate

O código de exemplo a seguir mostra como usar describe-certificate.

AWS CLI

Para obter informações sobre um certificado

O describe-certificate exemplo a seguir exibe os detalhes do certificado especificado.

```
aws iot describe-certificate \
  --certificate-
  id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

Saída:

```
{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
    "certificateId":
    "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
    "status": "ACTIVE",
    "certificatePem": "-----BEGIN CERTIFICATE-----
MIICiTEXAMPLEQD6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSBDEXAMPLE1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNEXAMPLEdBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BEXAMPLEz
b2xEXAMPLEYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8EXAMPLEZIHvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYEXAMPLEpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7EXAMPLEGBzZswY6786m86gpE
Ibb30hjZnzcvcQAARHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4
nUHVvXyUnEXAMPLE8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GEXAMPLE10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
    "ownedBy": "123456789012",
    "creationDate": 1541022751.983,
```

```

    "lastModifiedDate": 1541022751.983,
    "customerVersion": 1,
    "transferData": {},
    "generationId": "6974fbcd-2e61-4114-bc5e-4204cc79b045",
    "validity": {
      "notBefore": 1541022631.0,
      "notAfter": 2524607999.0
    }
  }
}

```

Para obter mais informações, consulte [DescribeCertificate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DescribeCertificate](#) na Referência de AWS CLI Comandos.

describe-custom-metric

O código de exemplo a seguir mostra como usar `describe-custom-metric`.

AWS CLI

Para obter informações sobre uma métrica personalizada do Device Defender

O `describe-custom-metric` exemplo a seguir obtém informações sobre uma métrica personalizada chamada `myCustomMetric`.

```

aws iot describe-custom-metric \
  --metric-name myCustomMetric

```

Saída:

```

{
  "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
}

```

Para obter mais informações, consulte [Métricas personalizadas](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [DescribeCustomMetric](#)na Referência de AWS CLI Comandos.

describe-default-authorizer

O código de exemplo a seguir mostra como usar `describe-default-authorizer`.

AWS CLI

Para obter informações sobre o autorizador personalizado padrão

O `describe-default-authorizer` exemplo a seguir exibe detalhes do autorizador personalizado padrão.

```
aws iot describe-default-authorizer
```

Saída:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

Para obter mais informações, consulte [DescribeDefaultAuthorizera](#) Referência de AWS IoT. API

- Para API obter detalhes, consulte [DescribeDefaultAuthorizerna](#) Referência de AWS CLI Comandos.

describe-dimension

O código de exemplo a seguir mostra como usar `describe-dimension`.

AWS CLI

Para obter informações sobre uma dimensão

O `describe-dimension` exemplo a seguir obtém informações sobre uma dimensão chamada `TopicFilterForAuthMessages`.

```
aws iot describe-dimension \
```

```
--name TopicFilterForAuthMessages
```

Saída:

```
{
  "name": "TopicFilterForAuthMessages",
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
  "type": "TOPIC_FILTER",
  "stringValues": [
    "device+/auth"
  ],
  "creationDate": 1578620223.255,
  "lastModifiedDate": 1578620223.255
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeDimension](#) na Referência de AWS CLI Comandos.

describe-domain-configuration

O código de exemplo a seguir mostra como usar `describe-domain-configuration`.

AWS CLI

Para descrever uma configuração de domínio

O `describe-domain-configuration` exemplo a seguir exibe detalhes sobre a configuração de domínio especificada.

```
aws iot describe-domain-configuration \
  --domain-configuration-name "additionalDataDomain"
```

Saída:

```
{
  "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  
}
```

Para obter mais informações, consulte [Endpoints configuráveis no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeDomainConfiguration](#) na Referência de AWS CLI Comandos.

describe-endpoint

O código de exemplo a seguir mostra como usar `describe-endpoint`.

AWS CLI

Exemplo 1: Para obter seu AWS endpoint atual

O `describe-endpoint` exemplo a seguir recupera o AWS endpoint padrão ao qual todos os comandos são aplicados.

```
aws iot describe-endpoint
```

Saída:

```
{  
  "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"  
}
```

Para obter mais informações, consulte o [DescribeEndpoint](#) Guia do desenvolvedor de AWS IoT.

Exemplo 2: Para obter seu ATS endpoint

O `describe-endpoint` exemplo a seguir recupera o endpoint Amazon Trust Services (ATS).

```
aws iot describe-endpoint \  
  --endpoint-type iot:Data-ATS
```

Saída:

```
{
  "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"
}
```

Para obter mais informações, consulte [Certificados X.509 e IoT AWS no Guia do desenvolvedor de AWS IoT](#).

- Para API obter detalhes, consulte [DescribeEndpoint](#) na Referência de AWS CLI Comandos.

describe-event-configurations

O código de exemplo a seguir mostra como usar `describe-event-configurations`.

AWS CLI

Para mostrar quais tipos de eventos são publicados

O `describe-event-configurations` exemplo a seguir lista a configuração que controla quais eventos são gerados quando algo é adicionado, atualizado ou excluído.

```
aws iot describe-event-configurations
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Mensagens de eventos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeEventConfigurations](#) na Referência de AWS CLI Comandos.

describe-index

O código de exemplo a seguir mostra como usar `describe-index`.

AWS CLI

Para recuperar o status atual do índice da coisa

O `describe-index` exemplo a seguir recupera o status atual do índice da coisa.

```
aws iot describe-index \
  --index-name "AWS_Things"
```

Saída:

```
{
```



```
"indexName": "AWS_Things",
"indexStatus": "ACTIVE",
"schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

Para obter mais informações, consulte [Managing Thing Indexing](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeIndex](#) na Referência de AWS CLI Comandos.

describe-job-execution

O código de exemplo a seguir mostra como usar `describe-job-execution`.

AWS CLI

Para obter detalhes de execução de um trabalho em um dispositivo

O `describe-job-execution` exemplo a seguir obtém detalhes de execução para o trabalho especificado.

```
aws iot describe-job-execution \
  --job-id "example-job-01" \
  --thing-name "MyRaspberryPi"
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeJobExecution](#) na Referência de AWS CLI Comandos.

describe-job

O código de exemplo a seguir mostra como usar `describe-job`.

AWS CLI

Para obter o status detalhado de um trabalho

O `describe-job` exemplo a seguir obtém o status detalhado do trabalho cujo ID é `example-job-01`.

```
aws iot describe-job \  
  --job-id "example-job-01"
```

Saída:

```
{  
  "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": {}
  }
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeJob](#) na Referência de AWS CLI Comandos.

describe-mitigation-action

O código de exemplo a seguir mostra como usar describe-mitigation-action.

AWS CLI

Para ver os detalhes de uma ação de mitigação definida

O describe-mitigation-action exemplo a seguir exibe detalhes da ação de mitigação especificada.

```
aws iot describe-mitigation-action \
  --action-name AddThingsToQuarantineGroupAction
```

Saída:

```
{
  "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"
  }

```

Para obter mais informações, consulte [DescribeMitigationAction \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeMitigationAction](#) na Referência de AWS CLI Comandos.

describe-provisioning-template-version

O código de exemplo a seguir mostra como usar `describe-provisioning-template-version`.

AWS CLI

Para descrever uma versão do modelo de aprovisionamento

O `describe-provisioning-template-version` exemplo a seguir descreve uma versão do modelo de aprovisionamento.

```

aws iot describe-provisioning-template-version \
  --template-name MyTestProvisioningTemplate \
  --version-id 1

```

Saída:

```

{
  "versionId": 1,
  "creationDate": 1589308310.574,
  "templateBody": "{
    \"Parameters\":{
      \"SerialNumber\":{
        \"Type\":\"String\"
      },
      \"AWS::IoT::Certificate::Id\":{
        \"Type\":\"String\"
      }
    },
    \"Resources\":{
      \"certificate\":{
        \"Properties\":{
          \"CertificateId\":{

```


describe-provisioning-template

O código de exemplo a seguir mostra como usar `describe-provisioning-template`.

AWS CLI

Para descrever um modelo de provisionamento

O `describe-provisioning-template` exemplo a seguir descreve um modelo de provisionamento.

```
aws iot describe-provisioning-template \
  --template-name MyTestProvisioningTemplate
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte Como [provisionar dispositivos que não têm certificados de dispositivo usando o provisionamento de frota no](#) Guia de desenvolvedores do IoT Core AWS .

- Para API obter detalhes, consulte [DescribeProvisioningTemplate](#) na Referência de AWS CLI Comandos.

describe-role-alias

O código de exemplo a seguir mostra como usar `describe-role-alias`.

AWS CLI

Para obter informações sobre um alias de função de AWS IoT

O `describe-role-alias` exemplo a seguir exibe detalhes do alias de função especificado.

```
aws iot describe-role-alias \  
  --role-alias LightBulbRole
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [DescribeRoleAlias](#) Referência de AWS IoT. API

- Para API obter detalhes, consulte [DescribeRoleAlias](#) na Referência de AWS CLI Comandos.

`describe-scheduled-audit`

O código de exemplo a seguir mostra como usar `describe-scheduled-audit`.

AWS CLI

Para obter informações sobre uma auditoria agendada

O `describe-scheduled-audit` exemplo a seguir obtém informações detalhadas sobre uma auditoria agendada AWS IOT do Device Defender chamada `AWSIoTDeviceDefenderDailyAudit`.

```
aws iot describe-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```


Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeScheduledAudit](#) na Referência de AWS CLI Comandos.

describe-security-profile

O código de exemplo a seguir mostra como usar `describe-security-profile`.

AWS CLI

Para obter informações sobre um perfil de segurança

O `describe-security-profile` exemplo a seguir obtém informações sobre o perfil de segurança do AWS IoT Device Defender chamado `PossibleIssue`.

```
aws iot describe-security-profile \
  --security-profile-name PossibleIssue
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeSecurityProfile](#) na Referência de AWS CLI Comandos.

describe-stream

O código de exemplo a seguir mostra como usar `describe-stream`.

AWS CLI

Para obter informações sobre um stream

O `describe-stream` exemplo a seguir exibe os detalhes sobre o fluxo especificado.

```
aws iot describe-stream \  
  --stream-id stream12345
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [DescribeStream](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DescribeStream](#) na Referência de AWS CLI Comandos.

describe-thing-group

O código de exemplo a seguir mostra como usar `describe-thing-group`.

AWS CLI

Para obter informações sobre um grupo de coisas

O `describe-thing-group` exemplo a seguir obtém informações sobre o grupo de coisas chamado `HalogenBulbs`.

```
aws iot describe-thing-group \  
  --thing-group-name HalogenBulbs
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [DescribeThingGroup](#) na Referência de AWS CLI Comandos.

describe-thing-type

O código de exemplo a seguir mostra como usar `describe-thing-type`.

AWS CLI

Para obter informações sobre um tipo de coisa

O `describe-thing-type` exemplo a seguir exibe informações sobre o tipo de item especificado definido em sua AWS conta.

```
aws iot describe-thing-type \  
  --thing-type-name "LightBulb"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [DescribeThingType](#) na Referência de AWS CLI Comandos.

describe-thing

O código de exemplo a seguir mostra como usar `describe-thing`.

AWS CLI

Para exibir informações detalhadas sobre uma coisa

O `describe-thing` exemplo a seguir exibe informações sobre uma coisa (dispositivo) definida no registro de AWS IoT da sua AWS conta.

como descrever a coisa --nome da coisa" MyLightBulb

Saída:

```
{
  "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
}
```

Para acessar mais informações, consulte [Como gerenciar objetos com o registro](#) no Guia do desenvolvedor do AWS IoT.

- Para API obter detalhes, consulte [DescribeThing](#) na Referência de AWS CLI Comandos.

detach-policy

O código de exemplo a seguir mostra como usar `detach-policy`.

AWS CLI

Exemplo 1: Para separar uma política de AWS IoT de um grupo de coisas

O `detach-policy` exemplo a seguir separa a política especificada de um grupo de coisas e, por extensão, de todas as coisas desse grupo e de qualquer um dos grupos secundários do grupo.

```
aws iot detach-policy \
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
  --policy-name "MyFirstGroup_Core-policy"
```

Este comando não produz saída.

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

Exemplo 2: Para separar uma política de AWS IoT de um certificado de dispositivo

O `detach-policy` exemplo a seguir separa a `TemperatureSensorPolicy` política de um certificado de dispositivo identificado por ARN.

```
aws iot detach-policy \  
  --policy-name TemperatureSensorPolicy \  
  --target arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DetachPolicy](#) na Referência de AWS CLI Comandos.

detach-security-profile

O código de exemplo a seguir mostra como usar detach-security-profile.

AWS CLI

Para desassociar um perfil de segurança de um alvo

O detach-security-profile exemplo a seguir remove a associação entre o Testprofile nome do perfil de segurança do AWS IoT Device Defender e o alvo de todas as coisas registradas.

```
aws iot detach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/registered-things"
```

Este comando não produz saída.

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DetachSecurityProfile](#) na Referência de AWS CLI Comandos.

detach-thing-principal

O código de exemplo a seguir mostra como usar detach-thing-principal.

AWS CLI

Para separar um certificado/diretor de uma coisa

O `detach-thing-principal` exemplo a seguir remove um certificado que representa um principal da coisa especificada.

```
aws iot detach-thing-principal \  
  --thing-name "MyLightBulb" \  
  --principal "arn:aws:iot:us-  
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

Este comando não produz saída.

Para acessar mais informações, consulte [Como gerenciar objetos com o registro](#) no Guia do desenvolvedor do AWS IoT.

- Para API obter detalhes, consulte [DetachThingPrincipal](#) na Referência de AWS CLI Comandos.

disable-topic-rule

O código de exemplo a seguir mostra como usar `disable-topic-rule`.

AWS CLI

Para desativar uma regra de tópico

O `disable-topic-rule` exemplo a seguir desativa a regra de tópico especificada.

```
aws iot disable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

Este comando não produz saída.

Para obter mais informações, consulte [Visualizando suas regras](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DisableTopicRule](#) na Referência de AWS CLI Comandos.

enable-topic-rule

O código de exemplo a seguir mostra como usar `enable-topic-rule`.

AWS CLI

Para habilitar uma regra de tópico

O `enable-topic-rule` exemplo a seguir ativa (ou reativa) a regra de tópico especificada.

```
aws iot enable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

Este comando não produz saída.

Para obter mais informações, consulte [Visualizando suas regras](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [EnableTopicRule](#) na Referência de AWS CLI Comandos.

`get-behavior-model-training-summaries`

O código de exemplo a seguir mostra como usar `get-behavior-model-training-summaries`.

AWS CLI

Para listar o status do modelo de treinamento do Perfil de Segurança ML Detect do Device Defender

O `get-behavior-model-training-summaries` exemplo a seguir lista o status de treinamento do modelo para os comportamentos configurados no Perfil de Segurança escolhido. Para cada comportamento, o nome, o status do modelo e a porcentagem de pontos de dados coletados são listados.

```
aws iot get-behavior-model-training-summaries \  
  --security-profile-name MySecuirtyProfileName
```

Saída:

```
{  
  "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": "MySecurityProfileName",
    "behaviorName": "Authorization_failures_ML_behavior",
    "modelStatus": "PENDING_BUILD",
    "datapointsCollectionPercentage": 0.0
  },
  {
    "securityProfileName": "MySecurityProfileName",
    "behaviorName": "Message_size_ML_behavior",
    "modelStatus": "PENDING_BUILD",
    "datapointsCollectionPercentage": 0.0
  },
  {
    "securityProfileName": "MySecurityProfileName",
    "behaviorName": "Connection_attempts_ML_behavior",
    "modelStatus": "PENDING_BUILD",
    "datapointsCollectionPercentage": 0.0
  },
  {
    "securityProfileName": "MySPNoAlerts",
    "behaviorName": "Disconnects_ML_behavior",
    "modelStatus": "PENDING_BUILD",
    "datapointsCollectionPercentage": 0.0
  }
]
}

```

Para obter mais informações, consulte [GetBehaviorModelTrainingSummaries \(Detectar comandos\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetBehaviorModelTrainingSummaries](#) na Referência de AWS CLI Comandos.

get-cardinality

O código de exemplo a seguir mostra como usar `get-cardinality`.

AWS CLI

Para retornar a contagem aproximada de valores exclusivos que correspondem à consulta

Você pode usar o script de configuração a seguir para criar 10 itens representando 10 sensores de temperatura. Cada coisa nova tem 3 atributos.

```
# 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
```

Exemplo de saída do script de configuração:

```
{
  "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"
}
```

O `get-cardinality` exemplo a seguir consulta os 10 sensores criados pelo script de configuração e retorna o número de racks que têm sensores de temperatura relatando valores de temperatura anormais. Se o valor da temperatura estiver abaixo de 60 ou acima de 80, o sensor de temperatura está em um estado anormal.

```
aws iot get-cardinality \
  --aggregation-field "attributes.rackId" \
  --query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

Saída:

```
{
  "cardinality": 2
}
```

Para obter mais informações, consulte Consultando dados agregados < <https://docs.aws.amazon.com/iot/latest/developerguide/index-aggregate.html> > no Guia do desenvolvedor de IoT.AWS

- Para API obter detalhes, consulte [GetCardinality](#) na Referência de AWS CLI Comandos.

get-effective-policies

O código de exemplo a seguir mostra como usar get-effective-policies.

AWS CLI

Para listar as políticas que afetam algo

O get-effective-policies exemplo a seguir lista as políticas que afetam a coisa especificada, incluindo políticas anexadas a qualquer grupo ao qual ela pertença.

```
aws iot get-effective-policies \
  --thing-name TemperatureSensor-001 \
  --principal arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Saída:

```
{
  "effectivePolicies": [
    {
      "policyName": "TemperatureSensorPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
      "policyDocument": "{
        \"Version\": \"2012-10-17\",
        \"Statement\": [
          {
            \"Effect\": \"Allow\",
            \"Action\": [
              \"iot:Publish\",
```


AWS CLI

Para obter a configuração de indexação da coisa

O `get-indexing-configuration` exemplo a seguir obtém os dados de configuração atuais para indexação de frotas de AWS IoT.

```
aws iot get-indexing-configuration
```

Saída:

```
{
  "thingIndexingConfiguration": {
    "thingIndexingMode": "OFF",
    "thingConnectivityIndexingMode": "OFF"
  },
  "thingGroupIndexingConfiguration": {
    "thingGroupIndexingMode": "OFF"
  }
}
```

Para obter mais informações, consulte [Managing Thing Indexing](#) no Guia de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [GetIndexingConfiguration](#) na Referência de AWS CLI Comandos.

get-job-document

O código de exemplo a seguir mostra como usar `get-job-document`.

AWS CLI

Para recuperar o documento para um trabalho

O `get-job-document` exemplo a seguir exibe detalhes sobre o documento do trabalho cujo ID é `example-job-01`.

```
aws iot get-job-document \
  --job-id "example-job-01"
```

Saída:

```
{
  "document": "\n{\n  \"operation\": \"customJob\", \n  \"otherInfo\":
  \"someValue\"\n}\n"
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetJobDocument](#) na Referência de AWS CLI Comandos.

get-logging-options

O código de exemplo a seguir mostra como usar `get-logging-options`.

AWS CLI

Para obter as opções de registro

O `get-logging-options` exemplo a seguir mostra as opções de registro atuais AWS da sua conta.

```
aws iot get-logging-options
```

Saída:

```
{
  "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",
  "logLevel": "ERROR"
}
```

Para obter mais informações, consulte o título no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetLoggingOptions](#) na Referência de AWS CLI Comandos.

get-ota-update

O código de exemplo a seguir mostra como usar `get-ota-update`.

AWS CLI

Para recuperar informações sobre uma atualização OTA

O `get-ota-update` exemplo a seguir exibe detalhes sobre a OTA atualização especificada.

```
aws iot get-ota-update \  
  --ota-update-id ota12345
```

Saída:

```
{  
  "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": {
      }
  }
}
```

Para obter mais informações, consulte [GetOTAUpdate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [GetOtaUpdate](#) na Referência de AWS CLI Comandos.

get-percentiles

O código de exemplo a seguir mostra como usar `get-percentiles`.

AWS CLI

Para agrupar os valores agregados que correspondem à consulta em agrupamentos de percentis

Você pode usar o script de configuração a seguir para criar 10 itens representando 10 sensores de temperatura. Cada coisa nova tem 1 atributo.

```
# 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
```

Exemplo de saída do script de configuração:

```
{
  "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"
}
```

O `get-percentiles` exemplo a seguir consulta os 10 sensores criados pelo script de configuração e retorna um valor para cada grupo de percentis especificado. O grupo de percentis "10" contém o valor do campo agregado que ocorre em aproximadamente 10% dos valores que correspondem à consulta. Na saída a seguir, `{"percent": 10.0, "value": 67,7}` significa que aproximadamente 10,0% dos valores de temperatura estão abaixo de 67,7.

```
aws iot get-percentiles \  
  --aggregation-field "attributes.temperature" \  
  --query-string "thingName:TempSensor*" \  
  --percentils 10 25 50 75 90
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Consultando dados agregados](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetPercentiles](#) na Referência de AWS CLI Comandos.

get-policy-version

O código de exemplo a seguir mostra como usar `get-policy-version`.

AWS CLI

Para obter informações sobre uma versão específica de uma política

O `get-policy-version` exemplo a seguir obtém informações sobre a primeira versão da política especificada.

```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id "1"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para desenvolvedores de AWS IoT](#).

- Para API obter detalhes, consulte [GetPolicyVersion](#) na Referência de AWS CLI Comandos.

get-policy

O código de exemplo a seguir mostra como usar `get-policy`.

AWS CLI

Para obter informações sobre a versão padrão de uma política

O `get-policy` exemplo a seguir recupera informações sobre a versão padrão da política especificada.

```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para desenvolvedores de AWS IoT](#).

- Para API obter detalhes, consulte [GetPolicy](#) na Referência de AWS CLI Comandos.

get-registration-code

O código de exemplo a seguir mostra como usar `get-registration-code`.

AWS CLI

Para obter o código de registro AWS específico da sua conta

O `get-registration-code` exemplo a seguir recupera o código AWS de registro específico da sua conta.

```
aws iot get-registration-code
```

Saída:

```
{
```

```
"registrationCode":  
"15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"  
}
```

Para obter mais informações, consulte [Use seu próprio certificado](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetRegistrationCode](#) na Referência de AWS CLI Comandos.

get-statistics

O código de exemplo a seguir mostra como usar `get-statistics`.

AWS CLI

Para pesquisar dados agregados no índice do dispositivo

O `get-statistics` exemplo a seguir retorna o número de itens que têm uma propriedade chamada `connectivity.connected` set to `false` (ou seja, o número de dispositivos que não estão conectados) na sombra do dispositivo.

```
aws iot get-statistics \  
  --index-name AWS_Things \  
  --query-string "connectivity.connected:false"
```

Saída:

```
{  
  "statistics": {  
    "count": 6  
  }  
}
```

Para obter mais informações, consulte [Como obter estatísticas sobre sua frota de dispositivos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetStatistics](#) na Referência de AWS CLI Comandos.

get-topic-rule-destination

O código de exemplo a seguir mostra como usar `get-topic-rule-destination`.

AWS CLI

Para obter um destino de regra de tópico

O `get-topic-rule-destination` exemplo a seguir obtém informações sobre o destino de uma regra de tópico.

```
aws iot get-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Saída:

```
{  
  "topicRuleDestination": {  
    "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
    "status": "DISABLED",  
    "httpUrlProperties": {  
      "confirmationUrl": "https://example.com"  
    }  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com destinos de regras de tópicos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetTopicRuleDestination](#) na Referência de AWS CLI Comandos.

get-topic-rule

O código de exemplo a seguir mostra como usar `get-topic-rule`.

AWS CLI

Para obter informações sobre uma regra

O `get-topic-rule` exemplo a seguir obtém informações sobre a regra especificada.

```
aws iot get-topic-rule \  
  --rule-name MyRPILowMoistureAlertRule
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Visualizando suas regras](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [GetTopicRule](#) na Referência de AWS CLI Comandos.

get-v2-logging-options

O código de exemplo a seguir mostra como usar `get-v2-logging-options`.

AWS CLI

Para listar as opções de registro atuais

O `get-v2-logging-options` exemplo a seguir lista as opções de registro atuais para AWS IoT.

```
aws iot get-v2-logging-options
```

Saída:

```
{
  "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",
  "defaultLogLevel": "WARN",
  "disableAllLogs": false
}
```

Para obter mais informações, consulte o título no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetV2 LoggingOptions](#) na Referência de AWS CLI Comandos.

list-active-violations

O código de exemplo a seguir mostra como usar `list-active-violations`.

AWS CLI

Para listar as violações ativas

O `list-active-violations` exemplo a seguir lista todas as violações do perfil de segurança especificado.

```
aws iot list-active-violations \
  --security-profile-name Testprofile
```

Saída:

```
{
  "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
}
]
}

```

- Para API obter detalhes, consulte [ListActiveViolations](#) na Referência de AWS CLI Comandos.

list-attached-policies

O código de exemplo a seguir mostra como usar `list-attached-policies`.

AWS CLI

Exemplo 1: Para listar as políticas anexadas a um grupo

O `list-attached-policies` exemplo a seguir lista as políticas anexadas ao grupo especificado.

```
aws iot list-attached-policies \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
```

Saída:

```
{  
  "policies": [  
    {  
      "policyName": "UpdateDeviceCertPolicy",  
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/  
UpdateDeviceCertPolicy"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

Exemplo 2: Para listar as políticas anexadas a um certificado de dispositivo

O `list-attached-policies` exemplo a seguir lista as políticas de AWS IoT anexadas ao certificado do dispositivo. O certificado é identificado por seu ARN.

```
aws iot list-attached-policies \  
  --target arn:aws:iot:us-  
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Saída:

```
{  
  "policies": [  
    {  
      "policyName": "TemperatureSensorPolicy",
```

```
        "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy"
      }
    ]
  }
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListAttachedPolicies](#) na Referência de AWS CLI Comandos.

list-audit-findings

O código de exemplo a seguir mostra como usar `list-audit-findings`.

AWS CLI

Exemplo 1: Para listar todas as descobertas de uma auditoria

O `list-audit-findings` exemplo a seguir lista todas as descobertas de uma auditoria do AWS IoT Device Defender com um ID de tarefa especificado.

```
aws iot list-audit-findings \
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Saída:

```
{
  "findings": []
}
```

Exemplo 2: Para listar as descobertas de um tipo de verificação de auditoria

O `list-audit-findings` exemplo a seguir mostra os resultados das auditorias do AWS IoT Device Defender realizadas entre 5 de junho de 2019 e 19 de junho de 2019, nas quais os dispositivos estão compartilhando um certificado de dispositivo. Ao especificar um nome de verificação, você deve fornecer um horário de início e término.

```
aws iot list-audit-findings \
  --check-name DEVICE_CERTIFICATE_SHARED_CHECK \
  --start-time 1559747125 \
  --end-time 1560962028
```

Saída:

```
{
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListAuditFindings](#) na Referência de AWS CLI Comandos.

list-audit-mitigation-actions-executions

O código de exemplo a seguir mostra como usar `list-audit-mitigation-actions-executions`.

AWS CLI

Para listar os detalhes da execução de uma ação de mitigação de auditoria

Uma tarefa de ação de mitigação de auditoria aplica uma ação de mitigação a uma ou mais descobertas de uma auditoria do IoT Device AWS Defender. O `list-audit-mitigation-actions-executions` exemplo a seguir lista os detalhes da tarefa de ação de mitigação com a descoberta especificada `taskId` e especificada.

```
aws iot list-audit-mitigation-actions-executions \  
  --task-id myActionsTaskId \  
  --finding-id 0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListAuditMitigationActionsExecutions \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListAuditMitigationActionsExecutions](#) na Referência de AWS CLI Comandos.

list-audit-mitigation-actions-tasks

O código de exemplo a seguir mostra como usar `list-audit-mitigation-actions-tasks`.

AWS CLI

Para listar as tarefas de ação de mitigação de auditoria

O `list-audit-mitigation-actions-tasks` exemplo a seguir lista as ações de mitigação que foram aplicadas às descobertas dentro do período de tempo especificado.

```
aws iot list-audit-mitigation-actions-tasks \
  --start-time 1594157400 \
  --end-time 1594157430
```

Saída:

```
{
  "tasks": [
    {
      "taskId": "0062f2d6-3999-488f-88c7-bef005414103",
      "startTime": "2020-07-07T14:30:15.172000-07:00",
      "taskStatus": "COMPLETED"
    }
  ]
}
```

Para obter mais informações, consulte [ListAuditMitigationActionsTasks \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListAuditMitigationActionsTasks](#) na Referência de AWS CLI Comandos.

list-audit-suppressions

O código de exemplo a seguir mostra como usar `list-audit-suppressions`.

AWS CLI

Para listar todas as supressões de descobertas de auditoria

O `list-audit-suppressions` exemplo a seguir lista todas as supressões ativas de descobertas de auditoria.

```
aws iot list-audit-suppressions
```

Saída:

```
{
  "suppressions": [
    {
      "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
      "resourceIdentifier": {
        "deviceCertificateId": "c7691e<shortened>"
      },
      "expirationDate": 1597881600.0,
      "suppressIndefinitely": false
    }
  ]
}
```

Para obter mais informações, consulte [Auditoria encontrando supressões no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [ListAuditSuppressions](#) na Referência de AWS CLI Comandos.

list-audit-tasks

O código de exemplo a seguir mostra como usar `list-audit-tasks`.

AWS CLI

Para listar todas as descobertas de uma auditoria

O `list-audit-tasks` exemplo a seguir lista as tarefas de auditoria executadas entre 5 de junho de 2019 e 12 de junho de 2019.

```
aws iot list-audit-tasks \
  --start-time 1559747125 \
  --end-time 1560357228
```

Saída:

```
{
  "tasks": [
    {
```

```
    "taskId": "a3aea009955e501a31b764abe1bebd3d",
    "taskStatus": "COMPLETED",
    "taskType": "ON_DEMAND_AUDIT_TASK"
  },
  {
    "taskId": "f76b4b5102b632cd9ae38a279c266da1",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "51d9967d9f9ff4d26529505f6d2c444a",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "eef61068b0eb03c456d746c5a26ee04",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "041c49557b7c7b04c079a49514b55589",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "82c7f2afac1562d18a4560be73998acc",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  },
  {
    "taskId": "ac9086b7222a2f5e2e17bb6fd30b3aeb",
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK"
  }
}
```

```
]
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListAuditTasks](#) na Referência de AWS CLI Comandos.

list-authorizers

O código de exemplo a seguir mostra como usar `list-authorizers`.

AWS CLI

Para listar seu autorizador personalizado

O `list-authorizers` exemplo a seguir lista os autorizadores personalizados em sua AWS conta.

```
aws iot list-authorizers
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [ListAuthorizers](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListAuthorizers](#) na Referência de AWS CLI Comandos.

list-billing-groups

O código de exemplo a seguir mostra como usar `list-billing-groups`.

AWS CLI

Para listar os grupos de cobrança da sua AWS conta e região

O `list-billing-groups` exemplo a seguir lista todos os grupos de cobrança definidos para sua AWS conta e AWS região.

```
aws iot list-billing-groups
```

Saída:

```
{
  "billingGroups": [
    {
      "groupName": "GroupOne",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
    }
  ]
}
```

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [ListBillingGroups](#) na Referência de AWS CLI Comandos.

list-ca-certificates

O código de exemplo a seguir mostra como usar `list-ca-certificates`.

AWS CLI

Para listar os certificados CA registrados em sua AWS conta

O `list-ca-certificates` exemplo a seguir lista os certificados CA registrados em sua AWS conta.

```
aws iot list-ca-certificates
```

Saída:

```
{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "status": "INACTIVE",
      "creationDate": 1569365372.053
    }
  ]
}
```

Para obter mais informações, consulte [Use seu próprio certificado](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListCaCertificates](#) na Referência de AWS CLI Comandos.

list-certificates-by-ca

O código de exemplo a seguir mostra como usar `list-certificates-by-ca`.

AWS CLI

Para listar todos os certificados de dispositivo assinados com um certificado CA

O `list-certificates-by-ca` exemplo a seguir lista todos os certificados de dispositivo em sua AWS conta que estão assinados com o certificado CA especificado.

```
aws iot list-certificates-by-ca \
  --ca-certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Saída:

```
{
  "certificates": [
```

```

    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId":
"488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "status": "ACTIVE",
      "creationDate": 1569363250.557
    }
  ]
}

```

Para obter mais informações, consulte [ListCertificatesByCA](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListCertificatesByCana](#) Referência de AWS CLI Comandos.

list-certificates

O código de exemplo a seguir mostra como usar `list-certificates`.

AWS CLI

Exemplo 1: Para listar os certificados registrados em sua AWS conta

O `list-certificates` exemplo a seguir lista todos os certificados registrados em sua conta. Se você tiver mais do que o limite de paginação padrão de 25, poderá usar o valor de `nextMarker` resposta desse comando e fornecê-lo ao próximo comando para obter o próximo lote de resultados. Repita até `nextMarker` retornar sem um valor.

```
aws iot list-certificates
```

Saída:

```

{
  "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
    }
  ]
}

```

- Para API obter detalhes, consulte [ListCertificates](#) na Referência de AWS CLI Comandos.

list-custom-metrics

O código de exemplo a seguir mostra como usar `list-custom-metrics`.

AWS CLI

Para listar suas métricas personalizadas

O `list-custom-metrics` exemplo a seguir lista todas as suas métricas personalizadas.

```
aws iot list-custom-metrics \  
  --region us-east-1
```

Saída:

```
{  
  "metricNames": [  
    "batteryPercentage"  
  ]  
}
```

Para obter mais informações, consulte [Métricas personalizadas](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [ListCustomMetrics](#) na Referência de AWS CLI Comandos.

list-dimensions

O código de exemplo a seguir mostra como usar `list-dimensions`.

AWS CLI

Para listar as dimensões da sua AWS conta

O `list-dimensions` exemplo a seguir lista todas as dimensões do AWS IoT Device Defender definidas na AWS sua conta.

```
aws iot list-dimensions
```

Saída:

```
{
```

```
"dimensionNames": [  
  "TopicFilterForAuthMessages",  
  "TopicFilterForActivityMessages"  
]  
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListDimensions](#) na Referência de AWS CLI Comandos.

list-domain-configurations

O código de exemplo a seguir mostra como usar `list-domain-configurations`.

AWS CLI

Para listar as configurações de domínio

O `list-domain-configurations` exemplo a seguir lista as configurações de domínio em sua AWS conta que têm o tipo de serviço especificado.

```
aws iot list-domain-configurations \  
  --service-type "DATA"
```

Saída:

```
{  
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Endpoints configuráveis no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListDomainConfigurations](#) na Referência de AWS CLI Comandos.

list-indices

O código de exemplo a seguir mostra como usar `list-indices`.

AWS CLI

Para listar os índices de pesquisa configurados

O `list-indices` exemplo a seguir lista todos os índices de pesquisa configurados em sua AWS conta. Se você não ativou a indexação de coisas, talvez não tenha nenhum índice.

```
aws iot list-indices
```

Saída:

```
{
  "indexNames": [
    "AWS_Things"
  ]
}
```

Para obter mais informações, consulte [Managing Thing Indexing](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListIndices](#) na Referência de AWS CLI Comandos.

list-job-executions-for-job

O código de exemplo a seguir mostra como usar `list-job-executions-for-job`.

AWS CLI

Para listar os trabalhos em sua AWS conta

O `list-job-executions-for-job` exemplo a seguir lista todas as execuções de um trabalho em sua AWS conta, especificado pelo `jobId`.

```
aws iot list-job-executions-for-job \  
  --job-id my-ota-job
```

Saída:

```
{  
  "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  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListJobExecutionsForJob](#) na Referência de AWS CLI Comandos.

list-job-executions-for-thing

O código de exemplo a seguir mostra como usar `list-job-executions-for-thing`.

AWS CLI

Para listar os trabalhos que foram executados para uma coisa

O `list-job-executions-for-thing` exemplo a seguir lista todos os trabalhos que foram executados para a coisa chamada `MyRaspberryPi`.

```
aws iot list-job-executions-for-thing \
  --thing-name "MyRaspberryPi"
```

Saída:

```
{
  "executionSummaries": [
    {
      "jobId": "example-job-01",
      "jobExecutionSummary": {
        "status": "QUEUED",
        "queuedAt": 1560787023.636,
        "lastUpdatedAt": 1560787023.636,
        "executionNumber": 1
      }
    }
  ]
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListJobExecutionsForThing](#) na Referência de AWS CLI Comandos.

list-jobs

O código de exemplo a seguir mostra como usar `list-jobs`.

AWS CLI

Para listar os trabalhos em sua AWS conta

O `list-jobs` exemplo a seguir lista todos os trabalhos em sua AWS conta, classificados pelo status do trabalho.

```
aws iot list-jobs
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListJobs](#) na Referência de AWS CLI Comandos.

list-mitigation-actions

O código de exemplo a seguir mostra como usar `list-mitigation-actions`.

AWS CLI

Para listar todas as ações de mitigação definidas

O `list-mitigation-actions` exemplo a seguir lista todas as ações de mitigação definidas para sua AWS conta e região. Para cada ação, o nome e ARN a data de criação são listados.

```
aws iot list-mitigation-actions
```

Saída:

```
{
```

```
"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"  
  }  
]  
}
```

Para obter mais informações, consulte [ListMitigationActions \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListMitigationActions](#) na Referência de AWS CLI Comandos.

list-mitigations-actions

O código de exemplo a seguir mostra como usar `list-mitigations-actions`.

AWS CLI

Para listar todas as ações de mitigação definidas

O `list-mitigations-actions` exemplo a seguir lista todas as ações de mitigação definidas para sua AWS conta e região. Para cada ação, o nome e ARN a data de criação são listados.

```
aws iot list-mitigation-actions
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [ListMitigationActions \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListMitigationsActions](#) na Referência de AWS CLI Comandos.

list-ota-updates

O código de exemplo a seguir mostra como usar `list-ota-updates`.

AWS CLI

Para listar OTA as atualizações da conta

O `list-ota-updates` exemplo a seguir lista as OTA atualizações disponíveis.

```
aws iot list-ota-updates
```

Saída:

```

{
  "otaUpdates": [
    {
      "otaUpdateId": "itsaupdate",
      "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/
itsaupdate",
      "creationDate": 1557863215.995
    }
  ]
}

```

Para obter mais informações, consulte [ListOTAUpdates](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListOtaUpdates](#) na Referência de AWS CLI Comandos.

list-outgoing-certificates

O código de exemplo a seguir mostra como usar `list-outgoing-certificates`.

AWS CLI

Para listar os certificados que estão sendo transferidos para uma AWS conta diferente

O `list-outgoing-certificates` exemplo a seguir lista todos os certificados de dispositivo que estão sendo transferidos para uma AWS conta diferente usando o `transfer-certificate` comando.

```
aws iot list-outgoing-certificates
```

Saída:

```
{
  "outgoingCertificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId":
"488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "transferredTo": "030714055129",
      "transferDate": 1569427780.441,
      "creationDate": 1569363250.557
    }
  ]
}
```

Para obter mais informações, consulte [ListOutgoingCertificates](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListOutgoingCertificates](#) na Referência de AWS CLI Comandos.

list-policies

O código de exemplo a seguir mostra como usar `list-policies`.

AWS CLI

Para listar as políticas definidas em sua AWS conta

O `list-policies` exemplo a seguir lista todas as políticas definidas em sua AWS conta.

```
aws iot list-policies
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para desenvolvedores de AWS IoT](#).

- Para API obter detalhes, consulte [ListPolicies](#) na Referência de AWS CLI Comandos.

`list-policy-versions`

O código de exemplo a seguir mostra como usar `list-policy-versions`.

AWS CLI

Exemplo 1: Para ver todas as versões de uma política

O `list-policy-versions` exemplo a seguir lista todas as versões da política especificada e suas datas de criação.

```
aws iot list-policy-versions \  
  --policy-name LightBulbPolicy
```

Saída:

```
{  
  "policyVersions": [  
    {  
      "versionId": "2",  
      "isDefaultVersion": true,  
      "createDate": 1559925941.924  
    },  
    {  
      "versionId": "1",  
      "isDefaultVersion": false,  
      "createDate": 1559925941.924  
    }  
  ]  
}
```

Para obter mais informações, consulte [Políticas de AWS IoT no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [ListPolicyVersions](#) na Referência de AWS CLI Comandos.

list-principal-things

O código de exemplo a seguir mostra como usar `list-principal-things`.

AWS CLI

Para listar as coisas anexadas a um diretor

O `list-principal-things` exemplo a seguir lista as coisas anexadas ao principal especificado por um ARN.

```
aws iot list-principal-things \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

Saída:

```
{
  "things": [
    "DeskLamp",
    "TableLamp"
  ]
}
```

Para obter mais informações, consulte [ListPrincipalThings](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListPrincipalThings](#) na Referência de AWS CLI Comandos.

list-provisioning-template-versions

O código de exemplo a seguir mostra como usar `list-provisioning-template-versions`.

AWS CLI

Para listar as versões do modelo de provisionamento

O `list-provisioning-template-versions` exemplo a seguir lista as versões disponíveis do modelo de provisionamento especificado.

```
aws iot list-provisioning-template-versions \
  --template-name "widget-template"
```

Saída:

```
{
  "versions": [
    {
      "versionId": 1,
      "creationDate": 1574800471.339,
      "isDefaultVersion": true
    },
    {
      "versionId": 2,
      "creationDate": 1574801192.317,
      "isDefaultVersion": false
    }
  ]
}
```

```
]
}
```

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor](#) do IoT Core AWS .

- Para API obter detalhes, consulte [ListProvisioningTemplateVersions](#) na Referência de AWS CLI Comandos.

list-provisioning-templates

O código de exemplo a seguir mostra como usar `list-provisioning-templates`.

AWS CLI

Para listar modelos de provisionamento

O `list-provisioning-templates` exemplo a seguir lista todos os modelos de provisionamento em sua AWS conta.

```
aws iot list-provisioning-templates
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor](#) do IoT Core AWS .

- Para API obter detalhes, consulte [ListProvisioningTemplates](#) na Referência de AWS CLI Comandos.

list-role-aliases

O código de exemplo a seguir mostra como usar `list-role-aliases`.

AWS CLI

Para listar os aliases da função de AWS IoT em sua conta AWS

O `list-role-aliases` exemplo a seguir lista os aliases da função de AWS IoT em sua conta.
AWS

```
aws iot list-role-aliases
```

Saída:

```
{
  "roleAliases": [
    "ResidentAlias",
    "ElectricianAlias"
  ]
}
```

Para obter mais informações, consulte [ListRoleAliases](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListRoleAliases](#) na Referência de AWS CLI Comandos.

list-scheduled-audits

O código de exemplo a seguir mostra como usar `list-scheduled-audits`.

AWS CLI

Para listar as auditorias agendadas para sua conta AWS

O `list-scheduled-audits` exemplo a seguir lista todas as auditorias agendadas para sua AWS conta.

```
aws iot list-scheduled-audits
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListScheduledAudits](#) na Referência de AWS CLI Comandos.

list-security-profiles-for-target

O código de exemplo a seguir mostra como usar `list-security-profiles-for-target`.

AWS CLI

Para listar os perfis de segurança anexados a um alvo

O `list-security-profiles-for-target` exemplo a seguir lista os perfis de segurança do AWS IoT Device Defender que estão conectados a dispositivos não registrados.

```
aws iot list-security-profiles-for-target \
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/  
unregistered-things"
```

Saída:


```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListSecurityProfilesForTarget](#) na Referência de AWS CLI Comandos.

list-security-profiles

O código de exemplo a seguir mostra como usar `list-security-profiles`.

AWS CLI

Para listar os perfis de segurança da sua AWS conta

O `list-security-profiles` exemplo a seguir lista todos os perfis de segurança do AWS IoT Device Defender definidos na AWS sua conta.

```
aws iot list-security-profiles
```

Saída:

```
{
  "securityProfileIdentifiers": [
    {
      "name": "Testprofile",
      "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
    }
  ]
}
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListSecurityProfiles](#) na Referência de AWS CLI Comandos.

list-streams

O código de exemplo a seguir mostra como usar `list-streams`.

AWS CLI

Para listar os streams na conta

O `list-streams` exemplo a seguir lista todos os streams em sua AWS conta.

```
aws iot list-streams
```

Saída:

```
{  
  "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."  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListStreams](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListStreams](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para exibir as tags e seus valores associados a um recurso

O `list-tags-for-resource` exemplo a seguir exibe as tags e os valores associados ao grupo de coisas `LightBulbs`.

```
aws iot list-tags-for-resource \  
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Saída:

```
{  
  "tags": [  
    {  
      "Key": "Assembly",  
      "Value": "Fact1NW"  
    },  
    {  
      "Key": "MyTag",  
      "Value": "777"  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [marcar seus recursos de AWS IoT](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-targets-for-policy

O código de exemplo a seguir mostra como usar `list-targets-for-policy`.

AWS CLI

Para listar os principais associados a uma política de AWS IoT

O `list-targets-for-policy` exemplo a seguir lista os certificados de dispositivo aos quais a política especificada está anexada.

```
aws iot list-targets-for-policy \  
  --policy-name UpdateDeviceCertPolicy
```

Saída:

```
{  
  "targets": [  
    "arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",  
    "arn:aws:iot:us-west-2:123456789012:cert/d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"  
  ]  
}
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListTargetsForPolicy](#) na Referência de AWS CLI Comandos.

`list-targets-for-security-profile`

O código de exemplo a seguir mostra como usar `list-targets-for-security-profile`.

AWS CLI

Para listar os alvos aos quais um perfil de segurança é aplicado

O `list-targets-for-security-profile` exemplo a seguir lista os alvos aos quais o PossibleIssue nome do perfil de segurança AWS IoT Device Defender é aplicado.

```
aws iot list-targets-for-security-profile \  
  --security-profile-name Testprofile
```

Saída:

```
{
  "securityProfileTargets": [
    {
      "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
    },
    {
      "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"
    }
  ]
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListTargetsForSecurityProfile](#) na Referência de AWS CLI Comandos.

list-thing-groups-for-thing

O código de exemplo a seguir mostra como usar `list-thing-groups-for-thing`.

AWS CLI

Para listar os grupos aos quais uma coisa pertence

O `list-thing-groups-for-thing` exemplo a seguir lista os grupos aos quais a coisa especificada pertence.

```
aws iot list-thing-groups-for-thing \
  --thing-name MyLightBulb
```

Saída:

```
{
  "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"
      }
    ]
  }
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListThingGroupsForThing](#) na Referência de AWS CLI Comandos.

list-thing-groups

O código de exemplo a seguir mostra como usar `list-thing-groups`.

AWS CLI

Para listar os grupos de coisas definidos em sua AWS conta

O `describe-thing-group` exemplo a seguir lista todos os grupos de itens definidos em sua AWS conta.

```
aws iot list-thing-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListThingGroups](#) na Referência de AWS CLI Comandos.

list-thing-principals

O código de exemplo a seguir mostra como usar `list-thing-principals`.

AWS CLI

Para listar os princípios associados a uma coisa

O `list-thing-principals` exemplo a seguir lista os principais (certificados X.509, IAM usuários, grupos, funções, identidades do Amazon Cognito ou identidades federadas) associados à coisa especificada.

```
aws iot list-thing-principals \  
  --thing-name MyRaspberryPi
```

Saída:

```
{  
  "principals": [  
    "arn:aws:iot:us-  
west-2:123456789012:cert/33475ac865079a5fffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"  
  ]  
}
```

Para obter mais informações, consulte [ListThingPrincipals](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListThingPrincipals](#) na Referência de AWS CLI Comandos.

list-thing-types

O código de exemplo a seguir mostra como usar `list-thing-types`.

AWS CLI

Para listar os tipos de coisas definidos

O `list-thing-types` exemplo a seguir exibe uma lista dos tipos de itens definidos em sua AWS conta.

```
aws iot list-thing-types
```

Saída:

```
{
  "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
      }
    }
  ]
}
```

Para obter mais informações, consulte [Tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListThingTypes](#) na Referência de AWS CLI Comandos.

list-things-in-billing-group

O código de exemplo a seguir mostra como usar `list-things-in-billing-group`.

AWS CLI

Para listar os itens em um grupo de cobrança

O `list-things-in-billing-group` exemplo a seguir lista os itens que estão no grupo de cobrança especificado.

```
aws iot list-things-in-billing-group \
  --billing-group-name GroupOne
```


Saída:

```
{
  "things": [
    "MyOtherLightBulb",
    "MyLightBulb"
  ]
}
```

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [ListThingsInBillingGroup](#) na Referência de AWS CLI Comandos.

list-things-in-thing-group

O código de exemplo a seguir mostra como usar `list-things-in-thing-group`.

AWS CLI

Para listar as coisas que pertencem a um grupo

O `list-things-in-thing-group` exemplo a seguir lista as coisas que pertencem ao grupo de coisas especificado.

```
aws iot list-things-in-thing-group \
  --thing-group-name LightBulbs
```

Saída:

```
{
  "things": [
    "MyLightBulb"
  ]
}
```

Para obter mais informações, consulte [Thing Groups](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListThingsInThingGroup](#) na Referência de AWS CLI Comandos.

list-things

O código de exemplo a seguir mostra como usar `list-things`.

AWS CLI

Exemplo 1: como listar todas os itens do registro

O `list-things` exemplo a seguir lista as coisas (dispositivos) que estão definidas no registro de AWS IoT da sua AWS conta.

```
aws iot list-things
```

Saída:

```
{
  "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
    }
  ]
}
```

Exemplo 2: como listar os itens definidos que possuem um atributo específico

O exemplo de `list-things` a seguir exibe uma lista dos itens que têm um atributo chamado `wattage`.

```
aws iot list-things \
  --attribute-name wattage
```

Saída:

```
{
  "things": [
    {
      "thingName": "MyLightBulb",
      "thingTypeName": "LightBulb",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
      "attributes": {
        "model": "123",
        "wattage": "75"
      }
    },
    {
      "thingName": "MyOtherLightBulb",
      "thingTypeName": "LightBulb",
      "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
      "attributes": {
        "model": "123",
        "wattage": "75"
      }
    }
  ]
}
```

```
    },  
    "version": 3  
  }  
]  
}
```

Para acessar mais informações, consulte [Como gerenciar objetos com o registro](#) no Guia do desenvolvedor do AWS IoT.

- Para API obter detalhes, consulte [ListThings](#) na Referência de AWS CLI Comandos.

list-topic-rule-destinations

O código de exemplo a seguir mostra como usar `list-topic-rule-destinations`.

AWS CLI

Para listar seus destinos de regras de tópicos

O `list-topic-rule-destinations` exemplo a seguir lista todos os destinos de regras de tópico que você definiu na AWS região atual.

```
aws iot list-topic-rule-destinations
```

Saída:

```
{  
  "destinationSummaries": [  
    {  
      "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
      "status": "ENABLED",  
      "httpUrlSummary": {  
        "confirmationUrl": "https://example.com"  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [trabalhar com destinos de regras de tópicos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListTopicRuleDestinations](#) na Referência de AWS CLI Comandos.

list-topic-rules

O código de exemplo a seguir mostra como usar `list-topic-rules`.

AWS CLI

Para listar suas regras

O `list-topic-rules` exemplo a seguir lista todas as regras que você definiu.

```
aws iot list-topic-rules
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Visualizando suas regras](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [ListTopicRules](#) na Referência de AWS CLI Comandos.

list-v2-logging-levels

O código de exemplo a seguir mostra como usar `list-v2-logging-levels`.

AWS CLI

Para listar os níveis de registro

O `list-v2-logging-levels` exemplo a seguir lista os níveis de registro configurados. Se os níveis de registro não foram definidos, um `NotConfiguredException` ocorre quando você executa esse comando.

```
aws iot list-v2-logging-levels
```

Saída:

```
{
  "logTargetConfigurations": [
    {
      "logTarget": {
        "targetType": "DEFAULT"
      },
      "logLevel": "ERROR"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListV2 LoggingLevels](#) na Referência de AWS CLI Comandos.

list-violation-events

O código de exemplo a seguir mostra como usar `list-violation-events`.

AWS CLI

Para listar as violações do perfil de segurança durante um período

O `list-violation-events` exemplo a seguir lista as violações que ocorreram entre 5 de junho de 2019 e 12 de junho de 2019 para todos os perfis de segurança do AWS IoT Device Defender da conta AWS atual AWS e da região.

```
aws iot list-violation-events \  
  --start-time 1559747125 \  
  --end-time 1560351925
```

Saída:

```
{  
  "violationEvents": [  
    {  
      "violationId": "174db59167fa474c80a652ad1583fd44",  
      "thingName": "iotconsole-1560269126751-1",  
      "securityProfileName": "Testprofile",  
      "behavior": {  
        "name": "Authorization",  
        "metric": "aws:num-authorization-failures",  
        "criteria": {  
          "comparisonOperator": "greater-than",  
          "value": {  
            "count": 10  
          },  
          "durationSeconds": 300,  
          "consecutiveDatapointsToAlarm": 1,  
          "consecutiveDatapointsToClear": 1  
        }  
      },  
      "metricValue": {  
        "count": 0  
      },  
      "violationEventType": "in-alarm",  
      "violationEventTime": 1560279000.0  
    },  
    {  
      "violationId": "c8a9466a093d3b7b35cd44ca58bdbbeab",  
      "thingName": "TvnQoEoU",  
      "securityProfileName": "Testprofile",  
      "behavior": {  
        "name": "CellularBandwidth",  
        "metric": "aws:message-byte-size",  
        "criteria": {  
          "comparisonOperator": "greater-than",  
          "value": {  
            "count": 128  
          }  
        },  
      },  
    }  
  ]  
}
```

```

        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    }
},
"metricValue": {
    "count": 110
},
"violationEventType": "in-alarm",
"violationEventTime": 1560276600.0
},
{
    "violationId": "74aa393adea02e6648f3ac362beed55e",
    "thingName": "iotconsole-1560269232412-2",
    "securityProfileName": "Testprofile",
    "behavior": {
        "name": "Authorization",
        "metric": "aws:num-authorization-failures",
        "criteria": {
            "comparisonOperator": "greater-than",
            "value": {
                "count": 10
            },
            "durationSeconds": 300,
            "consecutiveDatapointsToAlarm": 1,
            "consecutiveDatapointsToClear": 1
        }
    },
    "metricValue": {
        "count": 0
    },
    "violationEventType": "in-alarm",
    "violationEventTime": 1560276600.0
},
{
    "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
    "thingName": "TvnQoEoU",
    "securityProfileName": "Testprofile",
    "behavior": {
        "name": "Authorization",
        "metric": "aws:num-authorization-failures",
        "criteria": {
            "comparisonOperator": "greater-than",
            "value": {
                "count": 10
            }
        }
    }
}

```



```

        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    }
},
"metricValue": {
    "count": 0
},
"violationEventType": "in-alarm",
"violationEventTime": 1560276600.0
}
]
}

```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListViolationEvents](#) na Referência de AWS CLI Comandos.

register-ca-certificate

O código de exemplo a seguir mostra como usar `register-ca-certificate`.

AWS CLI

Para registrar um certificado de autoridade de certificação (CA)

O `register-ca-certificate` exemplo a seguir registra um certificado CA. O comando fornece o certificado CA e um certificado de verificação de chave que prova que você possui a chave privada associada ao certificado CA.

```

aws iot register-ca-certificate \
  --ca-certificate file://rootCA.pem \
  --verification-cert file://verificationCert.pem

```

Saída:

```

{
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",

```

```
"certificateId":  
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467"  
}
```

Para obter mais informações, consulte [R registerCACertificate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [RegisterCaCertificate](#) na Referência de AWS CLI Comandos.

register-certificate

O código de exemplo a seguir mostra como usar `register-certificate`.

AWS CLI

Para registrar um certificado de dispositivo autoassinado

O `register-certificate` exemplo a seguir registra o certificado do `deviceCert.pem` dispositivo assinado pelo certificado `rootCA.pem` CA. O certificado CA deve ser registrado antes de ser usado para registrar um certificado de dispositivo autoassinado. O certificado autoassinado deve ser assinado pelo mesmo certificado CA que você passa para esse comando.

```
aws iot register-certificate \  
--certificate-pem file://deviceCert.pem \  
--ca-certificate-pem file://rootCA.pem
```

Saída:

```
{  
  "certificateArn": "arn:aws:iot:us-  
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",  
  "certificateId":  
  "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"  
}
```

Para obter mais informações, consulte [RegisterCertificate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [RegisterCertificate](#) na Referência de AWS CLI Comandos.

register-thing

O código de exemplo a seguir mostra como usar `register-thing`.

AWS CLI

Para registrar uma coisa

O `register-thing` exemplo a seguir registra uma coisa usando um modelo de provisionamento.

```
aws iot register-thing \
  --template-body '{"Parameters":{"ThingName":
{"Type":"String"},"AWS::IoT::Certificate::Id":{"Type":"String"}}, "Resources":
{"certificate":{"Properties":{"CertificateId":
{"Ref":"AWS::IoT::Certificate::Id"},"Status":"Active"},"Type":"AWS::IoT::Certificate"},"poli
{"Properties":{"PolicyName":"MyIotPolicy"},"Type":"AWS::IoT::Policy"},"thing":
{"OverrideSettings":
{"AttributePayload":"MERGE","ThingGroups":"DO_NOTHING","ThingTypeName":"REPLACE"},"Propertie
{"AttributePayload":{},"ThingGroups":[],"ThingName":
{"Ref":"ThingName"},"ThingTypeName":"VirtualThings"},"Type":"AWS::IoT::Thing"}}}' \
  --parameters '{"ThingName":"Register-thing-
trial-1","AWS::IoT::Certificate::Id":"799a9ea048a1e6aea42b55EXAMPLEf8697b4bafcd77a318a3068e3
```

Saída:

```
{
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgIUYLk81I35cIppobpw
Hi0J2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLEkGA1UECwxQW1hem9uIFd1YiBTZXJ2aWNlcyBPPUFTYXpvbi
5jb20g\nSW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIwMDcyMzE2NDUw
\n0VoXDTQ5MTIzMT
IzNTk1OVowHjEcMBoGA1UEAwwTQVd0TIElVVCBZDZXJ0aWZpY2F0\nZTCCASIdQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBA071uADhdBajqTmgMV5\nnmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMackeX8Keb1I0TKzORI
RDXnyE
\n61V0wjgAsd0ku22rFxex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nnKqiuBJVNOGKTcQ
Hd1RN0rddwwu6kFJLeKDMEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nnGXu7u0W3hn6n1RN8qVoka0uW12p53xM7oHVz
Gf+cxKBx1b0hGkp6yCfTskUBm3Sp\nn9zLw35kiHXVm4EVpwn1nk6XcIGIkw8a/iy4pzmvuGAANY1/uU/
zgCjymw
ZT5S30\nnBV0CAwEAAaNgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR00BBYEF0VtvZ
9Aj2RYFnkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nnAQH/
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IB
```

```
AQCXCQcp0tubS5ft0sDMTcP/jNX
\nDHyaRxmjpSc2aCdmm7WX591TKWyAdxGAvqaDVWqTo0oXI7tZ8w7aINlGi5
pXnifx\n3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fe7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4sr0CKynS5wMJuQZ1HQ0nabVwnwK4Y0Mf1p
\np9+4susFUR9aT3BT1AcIwqSpzh1Khh4Iz7ND
kRn4amsUT210jg/z0010w+BTHcVQ\nJly8XDu0CWSu04q6SnaBzHmlySIajxuRTP/AdfRouP10Xe
+q1bP0BcvVvF
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"
  }
}
```

Para obter mais informações, consulte [Provisionamento por usuário confiável no Guia](#) para desenvolvedores do AWS IoT Core.

- Para API obter detalhes, consulte [RegisterThing](#) na Referência de AWS CLI Comandos.

reject-certificate-transfer

O código de exemplo a seguir mostra como usar `reject-certificate-transfer`.

AWS CLI

Para rejeitar uma transferência de certificado

O `reject-certificate-transfer` exemplo a seguir rejeita a transferência do certificado do dispositivo especificado de outra AWS conta.

```
aws iot reject-certificate-transfer \
  --certificate-
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78add5e605d630e05c7fc8
```

Este comando não produz saída.

Para obter mais informações, consulte [Transferir um certificado para outra conta](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [RejectCertificateTransfer](#) na Referência de AWS CLI Comandos.

remove-thing-from-billing-group

O código de exemplo a seguir mostra como usar `remove-thing-from-billing-group`.

AWS CLI

Para remover algo de um grupo de cobrança

O `remove-thing-from-billing-group` exemplo a seguir remove o item especificado de um grupo de cobrança.

```
aws iot remove-thing-from-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyOtherLightBulb
```

Este comando não produz saída.

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [RemoveThingFromBillingGroup](#) na Referência de AWS CLI Comandos.

remove-thing-from-thing-group

O código de exemplo a seguir mostra como usar `remove-thing-from-thing-group`.

AWS CLI

Para remover uma coisa de um grupo de coisas

O `remove-thing-from-thing-group` exemplo a seguir remove a coisa especificada de um grupo de coisas.

```
aws iot remove-thing-from-thing-group \  
  --thing-name bulb7 \  
  --thing-group-name DeadBulbs
```

Este comando não produz saída.

Para obter mais informações, consulte Thing Groups < <https://docs.aws.amazon.com/iot/latest/developerguide/thing-groups.html> > no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [RemoveThingFromThingGroup](#) na Referência de AWS CLI Comandos.

replace-topic-rule

O código de exemplo a seguir mostra como usar `replace-topic-rule`.

AWS CLI

Para atualizar a definição de regra de um tópico

O `replace-topic-rule` exemplo a seguir atualiza a regra especificada para enviar um SNS alerta quando as leituras do nível de umidade do solo estiverem muito baixas.

```
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\"}"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de uma regra de AWS IoT no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ReplaceTopicRule](#) na Referência de AWS CLI Comandos.

search-index

O código de exemplo a seguir mostra como usar `search-index`.

AWS CLI

Para consultar o índice da coisa

O `search-index` exemplo a seguir consulta o `AWS_Things` índice em busca de itens que têm um tipo de `LightBulb`.

```
aws iot search-index \  
  --index-name "AWS_Things" \  
  --query-string "thingTypeName:LightBulb"
```

Saída:

```
{  
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Managing Thing Indexing](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [SearchIndex](#) na Referência de AWS CLI Comandos.

set-default-authorizer

O código de exemplo a seguir mostra como usar `set-default-authorizer`.

AWS CLI

Para definir um autorizador padrão

O `set-default-authorizer` exemplo a seguir define o autorizador personalizado nomeado `CustomAuthorizer` como o autorizador padrão.

```
aws iot set-default-authorizer \  
  --authorizer-name CustomAuthorizer
```

Saída:

```
{  
  "authorizerName": "CustomAuthorizer",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer"  
}
```

Para obter mais informações, consulte [CreateDefaultAuthorizer](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [SetDefaultAuthorizer](#) na Referência de AWS CLI Comandos.

set-default-policy-version

O código de exemplo a seguir mostra como usar `set-default-policy-version`.

AWS CLI

Para definir a versão padrão de uma política

O `set-default-policy-version` exemplo a seguir define a versão padrão 2 para a política chamada `UpdateDeviceCertPolicy`.

```
aws iot set-default-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetDefaultPolicyVersion](#) na Referência de AWS CLI Comandos.

set-v2-logging-level

O código de exemplo a seguir mostra como usar `set-v2-logging-level`.

AWS CLI

Para definir o nível de registro para um grupo de coisas

O `set-v2-logging-level` exemplo a seguir define o nível de registro para registrar avisos para o grupo de coisas especificado.

```
aws iot set-v2-logging-level \  
  --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \  
  --log-level WARN
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetV2 LoggingLevel](#) na Referência de AWS CLI Comandos.

set-v2-logging-options

O código de exemplo a seguir mostra como usar `set-v2-logging-options`.

AWS CLI

Para definir as opções de registro

O `set-v2-logging-options` exemplo a seguir define o nível de verbosidade de registro padrão como `ERROR` e especifica o a ARN ser usado para registro.

```
aws iot set-v2-logging-options \  
  --default-log-level ERROR \  
  --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetV2 LoggingOptions](#) na Referência de AWS CLI Comandos.

start-audit-mitigation-actions-task

O código de exemplo a seguir mostra como usar `start-audit-mitigation-actions-task`.

AWS CLI

Aplicar uma ação de mitigação às conclusões de uma auditoria

O `start-audit-mitigation-actions-task` exemplo a seguir aplica a `ResetPolicyVersionAction` ação (que limpa a política) à descoberta única especificada.

```
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"
```

Saída:

```
{  
  "taskId": "myActionsTaskId"  
}
```

Para obter mais informações, consulte [StartAuditMitigationActionsTask \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [StartAuditMitigationActionsTask](#) na Referência de AWS CLI Comandos.

start-on-demand-audit-task

O código de exemplo a seguir mostra como usar `start-on-demand-audit-task`.

AWS CLI

Para iniciar uma auditoria imediatamente

O `start-on-demand-audit-task` exemplo a seguir inicia uma auditoria AWS do IoT Device Defender e executa três verificações de certificado.

```
aws iot start-on-demand-audit-task \  
  --target-check-  
  names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Saída:

```
{  
  "taskId": "a3aea009955e501a31b764abe1bebd3d"  
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [StartOnDemandAuditTask](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para especificar uma tag, chave e valor para um recurso

O `tag-resource` exemplo a seguir aplica a tag com uma chave `Assembly` e o valor `Fact1NW` ao grupo de coisas `LightBulbs`.

```
aws iot tag-resource \  
  --tags Key=Assembly,Value="Fact1NW" \  
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos de AWS IoT](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

test-authorization

O código de exemplo a seguir mostra como usar test-authorization.

AWS CLI

Para testar suas políticas AWS de IoT

O test-authorization exemplo a seguir testa as políticas de AWS IoT associadas ao principal especificado.

```
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
```

Saída:

```
{  
  "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": []
  }
]
}

```

Para obter mais informações, consulte [TestAuthorization](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [TestAuthorization](#) na Referência de AWS CLI Comandos.

test-invoke-authorizer

O código de exemplo a seguir mostra como usar `test-invoke-authorizer`.

AWS CLI

Para testar seu autorizador personalizado

O `test-invoke-authorizer` exemplo a seguir testa seu autorizador personalizado.

```

aws iot test-invoke-authorizer \
  --authorizer-name IoTAuthorizer \

```

```
--token allow \
--token-signature "mE0GvaHqy9nER/
FdgtJX5LXYEJ3b3vE7t1gEszc0TKGgLKWXtnPkb2AbKn0AZ81GyoN5dVtWDWVmr25m7+
+zjbYIMk2TBvyGXh0mvKFBPkdgyA43KL6SiZy0cTqLPMcQDsP7VX2rXr7CTowCxSNKphGXdQe0/
I5dQ+J06KUaHwCmupt0/MejKtaNwiia064j6wpr0AUwG5S1IYFuRd0X
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUSHEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcjxzGL02ExvyCUNctCV9dYLRGJj
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [TestInvokeAuthorizer](#) a Referência de AWS IoT. API

- Para API obter detalhes, consulte [TestInvokeAuthorizer](#) na Referência de AWS CLI Comandos.

transfer-certificate

O código de exemplo a seguir mostra como usar transfer-certificate.

AWS CLI

Para transferir um certificado de dispositivo para uma AWS conta diferente

O transfer-certificate exemplo a seguir transfere um certificado de dispositivo para outra AWS conta. O certificado e a AWS conta são identificados por ID.

```
aws iot transfer-certificate \
  --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \
  --target-aws-account 030714055129
```

Saída:

```
{
  "transferredCertificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}
```

Para obter mais informações, consulte [Transferir um certificado para outra conta](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [TransferCertificate](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma chave de tag de um recurso

O `untag-resource` exemplo a seguir remove a tag `MyTag` e seu valor do grupo de `coisasLightBulbs`.

```
command
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos de AWS IoT](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-account-audit-configuration

O código de exemplo a seguir mostra como usar `update-account-audit-configuration`.

AWS CLI

Exemplo 1: Para ativar as SNS notificações da Amazon para notificações de auditoria

O `update-account-audit-configuration` exemplo a seguir habilita SNS as notificações da Amazon para notificações de auditoria do AWS IoT Device Defender, especificando um alvo e a função usada para gravar nesse alvo.

```
aws iot update-account-audit-configuration \  
  --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-  
west-2:123456789012:ddauids\",roleArn=\"arn:aws:iam:123456789012:role/service-  
role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

Este comando não produz saída.

Exemplo 2: Para habilitar uma verificação de auditoria

O `update-account-audit-configuration` exemplo a seguir ativa a verificação `AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK` de auditoria do AWS IoT Device Defender chamada. Você não pode desativar uma verificação de auditoria se ela fizer parte de `targetCheckNames` uma ou mais auditorias agendadas para a AWS conta.

```
aws iot update-account-audit-configuration \  
  --audit-check-configurations  
  "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

Este comando não produz saída.

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateAccountAuditConfiguration](#) na Referência de AWS CLI Comandos.

update-audit-suppression

O código de exemplo a seguir mostra como usar `update-audit-suppression`.

AWS CLI

Para atualizar a supressão de uma descoberta de auditoria

O `update-audit-suppression` exemplo a seguir atualiza a data de expiração da supressão da constatação de auditoria para 2020-09-21.

```
aws iot update-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --resource-identifier deviceCertificateId=c7691e<shortened> \  
  --expiration-date 2020-09-21
```



```
--no-suppress-indefinitely \  
--expiration-date 2020-09-21
```

Este comando não produz saída.

Para obter mais informações, consulte [Auditoria encontrando supressões no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [UpdateAuditSuppression](#) na Referência de AWS CLI Comandos.

update-authorizer

O código de exemplo a seguir mostra como usar `update-authorizer`.

AWS CLI

Para atualizar um autorizador personalizado

O `update-authorizer` exemplo a seguir é o estado `CustomAuthorizer2` de dois `INACTIVE`.

```
aws iot update-authorizer \  
  --authorizer-name CustomAuthorizer2 \  
  --status INACTIVE
```

Saída:

```
{  
  "authorizerName": "CustomAuthorizer2",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer2"  
}
```

Para obter mais informações, consulte [UpdateAuthorizer](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [UpdateAuthorizer](#) na Referência de AWS CLI Comandos.

update-billing-group

O código de exemplo a seguir mostra como usar `update-billing-group`.

AWS CLI

Para atualizar as informações sobre um grupo de cobrança

O `update-billing-group` exemplo a seguir atualiza a descrição do grupo de cobrança especificado.

```
aws iot update-billing-group \  
  --billing-group-name GroupOne \  
  --billing-group-properties "billingGroupDescription=\"Primary bulb billing group  
\""
```

Saída:

```
{  
  "version": 2  
}
```

Para obter mais informações, consulte [Grupos de cobrança](#) no Guia para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [UpdateBillingGroup](#) na Referência de AWS CLI Comandos.

update-ca-certificate

O código de exemplo a seguir mostra como usar `update-ca-certificate`.

AWS CLI

Para atualizar um certificado de autoridade de certificação (CA)

O `update-ca-certificate` exemplo a seguir define o certificado CA especificado como ACTIVE status.

```
aws iot update-ca-certificate \  
  --certificate-  
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467 \  
  --new-status ACTIVE
```

Este comando não produz saída.

Para obter mais informações, consulte [UpdateCACertificate](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [UpdateCaCertificate](#) na Referência de AWS CLI Comandos.

update-certificate

O código de exemplo a seguir mostra como usar `update-certificate`.

AWS CLI

Para atualizar um certificado de dispositivo

O `update-certificate` exemplo a seguir define o certificado do dispositivo especificado como `INACTIVE` status.

```
aws iot update-certificate \  
  --certificate-  
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \  
  --new-status INACTIVE
```

Este comando não produz saída.

Para obter mais informações, consulte [UpdateCertificate](#) Referência de AWS IoT. API

- Para API obter detalhes, consulte [UpdateCertificate](#) na Referência de AWS CLI Comandos.

update-custom-metric

O código de exemplo a seguir mostra como usar `update-custom-metric`.

AWS CLI

Para atualizar uma métrica personalizada

O `update-custom-metric` exemplo a seguir atualiza uma métrica personalizada para ter uma `novadisplay-name`.

```
aws iot update-custom-metric \  
  --metric-name batteryPercentage \  
  --display-name 'remaining battery percentage on device' \  
  --region us-east-1
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Métricas personalizadas](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [UpdateCustomMetric](#) na Referência de AWS CLI Comandos.

update-dimension

O código de exemplo a seguir mostra como usar update-dimension.

AWS CLI

Para atualizar uma dimensão

O update-dimension exemplo a seguir atualiza uma dimensão.

```
aws iot update-dimension \
  --name TopicFilterForAuthMessages \
  --string-values device/${iot:ClientId}/auth
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Métricas de escopo em perfis de segurança usando dimensões](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [UpdateDimension](#) na Referência de AWS CLI Comandos.

update-domain-configuration

O código de exemplo a seguir mostra como usar `update-domain-configuration`.

AWS CLI

Para atualizar uma configuração de domínio

O `update-domain-configuration` exemplo a seguir desativa a configuração de domínio especificada.

```
aws iot update-domain-configuration \
  --domain-configuration-name "additionalDataDomain" \
  --domain-configuration-status "DISABLED"
```

Saída:

```
{
  "domainConfigurationName": "additionalDataDomain",
  "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

Para obter mais informações, consulte [Endpoints configuráveis no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateDomainConfiguration](#) na Referência de AWS CLI Comandos.

update-dynamic-thing-group

O código de exemplo a seguir mostra como usar `update-dynamic-thing-group`.

AWS CLI

Para atualizar um grupo dinâmico de coisas

O `update-dynamic-thing-group` exemplo a seguir atualiza o grupo dinâmico de coisas especificado. Ele fornece uma descrição e atualiza a string de consulta para alterar os critérios de associação ao grupo.

```
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"
```

Saída:

```
{  
  "version": 2  
}
```

Para obter mais informações, consulte [Dynamic Thing Groups](#) no Guia de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [UpdateDynamicThingGroup](#) na Referência de AWS CLI Comandos.

update-event-configurations

O código de exemplo a seguir mostra como usar `update-event-configurations`.

AWS CLI

Para mostrar quais tipos de eventos são publicados

O `update-event-configurations` exemplo a seguir atualiza a configuração para habilitar mensagens quando o certificado CA é adicionado, atualizado ou excluído.

```
aws iot update-event-configurations \  
  --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

Este comando não produz saída.

Para obter mais informações, consulte [Mensagens de eventos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateEventConfigurations](#) na Referência de AWS CLI Comandos.

update-indexing-configuration

O código de exemplo a seguir mostra como usar `update-indexing-configuration`.

AWS CLI

Para habilitar a indexação de coisas

O `update-indexing-configuration` exemplo a seguir permite que a indexação de coisas ofereça suporte à pesquisa de dados de registro, dados de sombra e status de conectividade de AWS coisas usando o índice `_Things`.

```
aws iot update-indexing-configuration
  --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

Este comando não produz saída.

Para obter mais informações, consulte [Managing Thing Indexing](#) no Guia de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [UpdateIndexingConfiguration](#) na Referência de AWS CLI Comandos.

update-job

O código de exemplo a seguir mostra como usar `update-job`.

AWS CLI

Para obter o status detalhado de um trabalho

O `update-job` exemplo a seguir obtém o status detalhado do trabalho cujo ID é `example-job-01`.

```
aws iot describe-job \
  --job-id "example-job-01"
```

Saída:

```
{
  "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": {}
  }
}
```

Para obter mais informações, consulte [Criação e gerenciamento de trabalhos \(CLI\)](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateJob](#) na Referência de AWS CLI Comandos.

update-mitigation-action

O código de exemplo a seguir mostra como usar `update-mitigation-action`.

AWS CLI

Para atualizar uma ação de mitigação

O `update-mitigation-action` exemplo a seguir atualiza a ação de mitigação especificada chamada `AddThingsToQuarantineGroupAction`, altera o nome do grupo de coisas e

define `overrideDynamicGroups` como. `false` Você pode verificar suas alterações usando o `describe-mitigation-action` comando.

```
aws iot update-mitigation-action \  
  --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",  
  \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":  
  [\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"
```

Saída:

```
{  
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/  
  AddThingsToQuarantineGroupAction",  
  "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"  
}
```

Para obter mais informações, consulte [UpdateMitigationAction \(Comandos de ação de mitigação\) no Guia](#) do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateMitigationAction](#) na Referência de AWS CLI Comandos.

update-provisioning-template

O código de exemplo a seguir mostra como usar `update-provisioning-template`.

AWS CLI

Para atualizar um modelo de aprovisionamento

O `update-provisioning-template` exemplo a seguir modifica a descrição e a função `arn` do modelo de provisionamento especificado e ativa o modelo.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS IoT Secure Tunneling no Guia do Desenvolvedor](#) do IoT Core AWS .

- Para API obter detalhes, consulte [UpdateProvisioningTemplate](#) na Referência de AWS CLI Comandos.

update-role-alias

O código de exemplo a seguir mostra como usar `update-role-alias`.

AWS CLI

Para atualizar um alias de função

O `update-role-alias` exemplo a seguir atualiza o alias da `LightBulbRole` função.

```
aws iot update-role-alias \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Saída:

```
{  
  "roleAlias": "LightBulbRole",  
  "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"  
}
```

Para obter mais informações, consulte [UpdateRoleAlias](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [UpdateRoleAlias](#) na Referência de AWS CLI Comandos.

update-scheduled-audit

O código de exemplo a seguir mostra como usar `update-scheduled-audit`.

AWS CLI

Para atualizar uma definição de auditoria agendada

O `update-scheduled-audit` exemplo a seguir altera os nomes das verificações de destino para uma auditoria agendada AWS do 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
```

Saída:

```
{
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

Para obter mais informações, consulte [Comandos de auditoria](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateScheduledAudit](#) na Referência de AWS CLI Comandos.

update-security-profile

O código de exemplo a seguir mostra como usar update-security-profile.

AWS CLI

Para alterar um perfil de segurança

O update-security-profile exemplo a seguir atualiza a descrição e os comportamentos de um perfil de segurança AWS do 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}]]"
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateSecurityProfile](#) na Referência de AWS CLI Comandos.

update-stream

O código de exemplo a seguir mostra como usar `update-stream`.

AWS CLI

Para atualizar um stream

O `update-stream` exemplo a seguir atualiza um stream existente. A versão do stream é incrementada em um.

```
aws iot update-stream \  
  --cli-input-json file://update-stream.json
```

Conteúdo de `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"  
}
```

Saída:

```
{  
  "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  
}
```

Para obter mais informações, consulte [UpdateStream](#) a Referência de AWS IoT. API


```
--thing-groups-to-add "replaceableItems" \  
--thing-groups-to-remove "DeadBulbs"
```

Este comando não produz saída.

Para obter mais informações, consulte [Thing Groups](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateThingGroupsForThing](#) na Referência de AWS CLI Comandos.

update-thing

O código de exemplo a seguir mostra como usar update-thing.

AWS CLI

Para associar uma coisa a um tipo de coisa

O update-thing exemplo a seguir associa uma coisa no registro de AWS IoT a um tipo de coisa. Ao fazer a associação, você fornece valores para os atributos definidos pelo tipo de coisa.

```
aws iot update-thing \  
  --thing-name "MyOtherLightBulb" \  
  --thing-type-name "LightBulb" \  
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

Esse comando não produz saída. Use o describe-thing comando para ver o resultado.

Para obter mais informações, consulte [Tipos de coisas](#) no Guia AWS para desenvolvedores de IoT.

- Para API obter detalhes, consulte [UpdateThing](#) na Referência de AWS CLI Comandos.

update-topic-rule-destination

O código de exemplo a seguir mostra como usar update-topic-rule-destination.

AWS CLI

Exemplo 1: Para habilitar um destino de regra de tópico

O update-topic-rule-destination exemplo a seguir ativa o tráfego para um destino de regra de tópico.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status ENABLED
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando um destino de regra de tópico](#) no Guia do desenvolvedor de AWS IoT.

Exemplo 2: Para desativar o destino de uma regra de tópico

O `update-topic-rule-destination` exemplo a seguir desativa o tráfego para um destino de regra de tópico.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status DISABLED
```

Este comando não produz saída.

Para obter mais informações, consulte Como [desativar o destino de uma regra de tópico](#) no Guia do desenvolvedor de AWS IoT.

Exemplo 3: Para enviar uma nova mensagem de confirmação

O `update-topic-rule-destination` exemplo a seguir envia uma nova mensagem de confirmação para um destino de regra de tópico.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status IN_PROGRESS
```

Este comando não produz saída.

Para obter mais informações, consulte [Enviar uma nova mensagem de confirmação](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateTopicRuleDestination](#) na Referência de AWS CLI Comandos.

validate-security-profile-behaviors

O código de exemplo a seguir mostra como usar `validate-security-profile-behaviors`.

AWS CLI

Exemplo 1: Para validar os parâmetros de comportamento de um perfil de segurança

O `validate-security-profile-behaviors` exemplo a seguir valida um conjunto bem formado e correto de comportamentos para um perfil de segurança do 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":1,"consecutiveDatapointsToClear":1}}]"
```

Saída:

```
{
  "valid": true,
  "validationErrors": []
}
```

Exemplo 2: Para validar parâmetros de comportamento incorretos para um perfil de segurança

O `validate-security-profile-behaviors` exemplo a seguir valida um conjunto de comportamentos que contém um erro para um perfil de segurança do 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}}]"
```

Saída:

```
{
  "valid": false,
  "validationErrors": [
    {
      "errorMessage": "Behavior Authorization is malformed.
consecutiveDatapointsToAlarm 100000 should be in range[1,10]"
    }
  ]
}
```

Para obter mais informações, consulte [Detectar comandos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ValidateSecurityProfileBehaviors](#) na Referência de AWS CLI Comandos.

AWS IoT 1-Click Exemplos de dispositivos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT 1-Click Devices.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

claim-devices-by-claim-code

O código de exemplo a seguir mostra como usar `claim-devices-by-claim-code`.

AWS CLI

Para reivindicar um ou mais dispositivos AWS IoT 1-Click usando um código de reivindicação

O `claim-devices-by-claim-code` exemplo a seguir reivindica o dispositivo AWS IoT 1-Click especificado usando um código de solicitação (em vez de uma ID do dispositivo).

```
aws iot1click-devices claim-devices-by-claim-code \  
  --claim-code C-123EXAMPLE
```

Saída:

```
{  
  "Total": 9  
  "ClaimCode": "C-123EXAMPLE"  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ClaimDevicesByClaimCode](#) na Referência de AWS CLI Comandos.

describe-device

O código de exemplo a seguir mostra como usar `describe-device`.

AWS CLI

Para descrever um dispositivo

O `describe-device` exemplo a seguir descreve o dispositivo especificado.

```
aws iot1click-devices describe-device \  
  --device-id G030PM0123456789
```

Saída:

```
{  
  "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": {}
  }
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [DescribeDevice](#) na Referência de AWS CLI Comandos.

finalize-device-claim

O código de exemplo a seguir mostra como usar `finalize-device-claim`.

AWS CLI

Para finalizar uma solicitação de reclamação para um dispositivo IoT AWS 1-Click usando uma ID de dispositivo

O `finalize-device-claim` exemplo a seguir finaliza uma solicitação de solicitação para o dispositivo IoT AWS 1-Click especificado usando uma ID do dispositivo (em vez de um código de solicitação).

```
aws iot1click-devices finalize-device-claim \
  --device-id G030PM0123456789
```

Saída:

```
{
  "State": "CLAIMED"
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [FinalizeDeviceClaim](#) na Referência de AWS CLI Comandos.

get-device-methods

O código de exemplo a seguir mostra como usar `get-device-methods`.

AWS CLI

Para listar os métodos disponíveis para um dispositivo

O `get-device-methods` exemplo a seguir lista os métodos disponíveis para um dispositivo.

```
aws iot1click-devices get-device-methods \  
  --device-id G030PM0123456789
```

Saída:

```
{  
  "DeviceMethods": [  
    {  
      "MethodName": "getDeviceHealthParameters"  
    },  
    {  
      "MethodName": "setDeviceHealthMonitorCallback"  
    },  
    {  
      "MethodName": "getDeviceHealthMonitorCallback"  
    },  
    {  
      "MethodName": "setOnClickCallback"  
    },  
    {  
      "MethodName": "getOnClickCallback"  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [GetDeviceMethods](#) na Referência de AWS CLI Comandos.

initiate-device-claim

O código de exemplo a seguir mostra como usar `initiate-device-claim`.

AWS CLI

Para iniciar uma solicitação de reclamação para um dispositivo IoT AWS 1-Click usando uma ID de dispositivo

O `initiate-device-claim` exemplo a seguir inicia uma solicitação de solicitação para o dispositivo IoT AWS 1-Click especificado usando uma ID do dispositivo (em vez de um código de solicitação).

```
aws iot1click-devices initiate-device-claim \  
  --device-id G030PM0123456789
```

Saída:

```
{  
  "State": "CLAIM_INITIATED"  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [InitiateDeviceClaim](#) na Referência de AWS CLI Comandos.

invoke-device-method

O código de exemplo a seguir mostra como usar `invoke-device-method`.

AWS CLI

Para invocar um método de dispositivo em um dispositivo

O `invoke-device-method` exemplo a seguir invoca o método especificado em um dispositivo.

```
aws iot1click-devices invoke-device-method \  
  --cli-input-json file://invoke-device-method.json
```

Conteúdo de `invoke-device-method.json`:

```
{  
  "DeviceId": "G030PM0123456789",
```

```
"DeviceMethod": {
  "DeviceType": "device",
  "MethodName": "getDeviceHealthParameters"
}
}
```

Saída:

```
{
  "DeviceMethodResponse": "{\"remainingLife\": 99.8}"
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [InvokeDeviceMethod](#) na Referência de AWS CLI Comandos.

list-device-events

O código de exemplo a seguir mostra como usar `list-device-events`.

AWS CLI

Para listar os eventos de um dispositivo em um intervalo de tempo especificado

O `list-device-events` exemplo a seguir lista os eventos do dispositivo especificado para o intervalo de tempo especificado.

```
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
```

Saída:

```
{
  "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}"
  }
]
}

```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ListDeviceEvents](#) na Referência de AWS CLI Comandos.

list-devices

O código de exemplo a seguir mostra como usar `list-devices`.

AWS CLI

Para listar os dispositivos de um tipo especificado

O `list-devices` exemplo a seguir lista os dispositivos de um tipo especificado.

```
aws iot1click-devices list-devices \
  --device-type button
```

Este comando não produz saída.

Saída:


```
{
  "Devices": [
    {
      "remainingLife": 99.9,
      "attributes": {
        "arn": "arn:aws:iot1click:us-west-2:123456789012:devices/
G030PM0123456789",
        "type": "button",
        "deviceId": "G030PM0123456789",
        "enabled": false
      }
    }
  ]
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ListDevices](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um dispositivo

O `list-tags-for-resource` exemplo a seguir lista as tags do dispositivo especificado.

```
aws iot1click-devices list-tags-for-resource \
  --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/
G030PM0123456789"
```

Saída:

```
{
  "Tags": {
    "Driver Phone": "123-555-0199",
    "Driver": "Jorge Souza"
  }
}
```

```
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um AWS recurso do dispositivo

O `tag-resource` exemplo a seguir adiciona duas tags ao recurso especificado.

```
aws iot1click-devices tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

Conteúdo de `devices-tag-resource.json`:

```
{  
  "ResourceArn": "arn:aws:iot1click:us-west-2:123456789012:devices/  
G030PM0123456789",  
  "Tags": {  
    "Driver": "Jorge Souza",  
    "Driver Phone": "123-555-0199"  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

unclaim-device

O código de exemplo a seguir mostra como usar `unclaim-device`.

AWS CLI

Para cancelar (cancelar o registro) de um dispositivo da sua conta AWS

O `unclaim-device` exemplo a seguir não reivindica (cancela o registro) do dispositivo especificado da sua conta. AWS

```
aws iot1click-devices unclaim-device \  
  --device-id G030PM0123456789
```

Saída:

```
{  
  "State": "UNCLAIMED"  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [UnclaimDevice](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um AWS recurso do dispositivo

O `untag-resource` exemplo a seguir remove as tags com os nomes `Driver Phone` e `Driver` do recurso de dispositivo especificado.

```
aws iot1click-devices untag-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters" \  
  --tag-keys "Driver Phone" "Driver"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-device-state

O código de exemplo a seguir mostra como usar `update-device-state`.

AWS CLI

Para atualizar o estado ``habilitado`` de um dispositivo

O seguinte `update-device-state` define o estado do dispositivo especificado como `enabled`.

```
aws iot1click-devices update-device-state \  
  --device-id G030PM0123456789 \  
  --enabled
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [UpdateDeviceState](#) na Referência de AWS CLI Comandos.

AWS IoT 1-Click Exemplos de projetos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT 1-Click Projects.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-device-with-placement

O código de exemplo a seguir mostra como usar `associate-device-with-placement`.

AWS CLI

Para associar um dispositivo de AWS IoT 1-Click a um posicionamento existente

O `associate-device-with-placement` exemplo a seguir associa o dispositivo IoT AWS 1-Click especificado a um posicionamento existente.

```
aws iot1click-projects associate-device-with-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request \  
  --device-id G030PM0123456789
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [AssociateDeviceWithPlacement](#) na Referência de AWS CLI Comandos.

`create-placement`

O código de exemplo a seguir mostra como usar `create-placement`.

AWS CLI

Para criar um posicionamento de AWS IoT 1-Click para um projeto

O `create-placement` exemplo a seguir cria um posicionamento de AWS IoT 1-Click para o projeto especificado.

```
aws iot1click-projects create-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --attributes '{"location": "123 Any Street Anytown, USA 10001", "phone":  
  "123-456-7890"}'
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [CreatePlacement](#) na Referência de AWS CLI Comandos.

create-project

O código de exemplo a seguir mostra como usar `create-project`.

AWS CLI

Para criar um projeto de AWS IoT 1-Click para zero ou mais colocações

O `create-project` exemplo a seguir cria um projeto de AWS IoT 1-Click para um posicionamento.

```
aws iot1click-projects create-project -- file: //create-project.json cli-input-json
```

Conteúdo de `create-project.json`:

```
{
  "projectName": "AnytownDumpsters",
  "description": "All dumpsters in the Anytown region.",
  "placementTemplate": {
    "defaultAttributes": {
      "City" : "Anytown"
    },
    "deviceTemplates": {
      "empty-dumpster-request" : {
        "deviceType": "button"
      }
    }
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [CreateProject](#) na Referência de AWS CLI Comandos.

delete-placement

O código de exemplo a seguir mostra como usar `delete-placement`.

AWS CLI

Para excluir um posicionamento de um projeto

O `delete-placement` exemplo a seguir exclui o posicionamento especificado de um projeto.

```
aws iot1click-projects delete-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [DeletePlacement](#) na Referência de AWS CLI Comandos.

`delete-project`

O código de exemplo a seguir mostra como usar `delete-project`.

AWS CLI

Para excluir um projeto da sua AWS conta

O `delete-project` exemplo a seguir exclui o projeto especificado da sua AWS conta.

```
aws iot1click-projects delete-project \  
  --project-name AnytownDumpsters
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [DeleteProject](#) na Referência de AWS CLI Comandos.

`describe-placement`

O código de exemplo a seguir mostra como usar `describe-placement`.

AWS CLI

Para descrever uma colocação para um projeto

O `describe-placement` exemplo a seguir descreve um posicionamento para o projeto especificado.

```
aws iot1click-projects describe-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Saída:

```
{  
  "placement": {  
    "projectName": "AnytownDumpsters",  
    "placementName": "customer217",  
    "attributes": {  
      "phone": "123-555-0110",  
      "location": "123 Any Street Anytown, USA 10001"  
    },  
    "createdDate": 1563488454,  
    "updatedAt": 1563488454  
  }  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [DescribePlacement](#) na Referência de AWS CLI Comandos.

`describe-project`

O código de exemplo a seguir mostra como usar `describe-project`.

AWS CLI

Para descrever um projeto de AWS IoT 1-Click

O `describe-project` exemplo a seguir descreve o projeto de AWS IoT 1-Click especificado.

```
aws iot1click-projects describe-project \  
  --project-name AnytownDumpsters
```



```
--project-name AnytownDumpsters
```

Saída:

```
{
  "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": {}
  }
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [DescribeProject](#) na Referência de AWS CLI Comandos.

disassociate-device-from-placement

O código de exemplo a seguir mostra como usar `disassociate-device-from-placement`.

AWS CLI

Para desassociar um dispositivo de um posicionamento

O `disassociate-device-from-placement` exemplo a seguir dissocia o dispositivo especificado de um posicionamento.

```
aws iot1click-projects disassociate-device-from-placement \
```

```
--project-name AnytownDumpsters \  
--placement-name customer217 \  
--device-template-name empty-dumpster-request
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [DisassociateDeviceFromPlacement](#) na Referência de AWS CLI Comandos.

get-devices-in-placement

O código de exemplo a seguir mostra como usar `get-devices-in-placement`.

AWS CLI

Para listar todos os dispositivos em um posicionamento contido em um projeto

O `get-devices-in-placement` exemplo a seguir lista todos os dispositivos em um posicionamento especificado contido no projeto especificado.

```
aws iot1click-projects get-devices-in-placement \  
--project-name AnytownDumpsters \  
--placement-name customer217
```

Saída:

```
{  
  "devices": {  
    "empty-dumpster-request": "G030PM0123456789"  
  }  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [GetDevicesInPlacement](#) na Referência de AWS CLI Comandos.

list-placements

O código de exemplo a seguir mostra como usar `list-placements`.

AWS CLI

Para listar todos os AWS posicionamentos do IoT 1-Click para um projeto

O `list-placements` exemplo a seguir lista todos os posicionamentos AWS do IoT 1-Click para o projeto especificado.

```
aws iot1click-projects list-placements \  
  --project-name AnytownDumpsters
```

Saída:

```
{  
  "placements": [  
    {  
      "projectName": "AnytownDumpsters",  
      "placementName": "customer217",  
      "createdDate": 1563488454,  
      "updatedAt": 1563488454  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ListPlacements](#) na Referência de AWS CLI Comandos.

list-projects

O código de exemplo a seguir mostra como usar `list-projects`.

AWS CLI

Para listar todos os projetos AWS do IoT 1-Click

O `list-projects` exemplo a seguir lista todos os projetos de AWS IoT 1-Click em sua conta.

```
aws iot1click-projects list-projects
```

Saída:

```
{
  "projects": [
    {
      "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/
AnytownDumpsters",
      "projectName": "AnytownDumpsters",
      "createdDate": 1563483100,
      "updatedDate": 1563483100,
      "tags": {}
    }
  ]
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ListProjects](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso do projeto

O `list-tags-for-resource` exemplo a seguir lista as tags do recurso de projeto especificado.

```
aws iot1click-projects list-tags-for-resource \
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters"
```

Saída:

```
{
```

```
"tags": {
  "Manager": "Li Juan",
  "Account": "45215"
}
}
```

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para adicionar tags a um recurso do projeto

O tag-resource exemplo a seguir adiciona duas tags ao recurso de projeto especificado.

```
aws iot1click-projects tag-resource \
  --cli-input-json file://devices-tag-resource.json
```

Conteúdo de devices-tag-resource.json:

```
{
  "resourceArn": "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters",
  "tags": {
    "Account": "45215",
    "Manager": "Li Juan"
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um recurso do projeto

O `untag-resource` exemplo a seguir remove a tag com o nome `Manager` da chave do projeto especificado.

```
aws iot1click-projects untag-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters" \  
  --tag-keys "Manager"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-placement

O código de exemplo a seguir mostra como usar `update-placement`.

AWS CLI

Para atualizar os pares de valores-chave de “atributos” de um posicionamento

O `update-placement` exemplo a seguir atualiza os pares de valores-chave “atributos” de um posicionamento.

```
aws iot1click-projects update-placement \  
  --cli-input-json file://update-placement.json
```

Conteúdo de `update-placement.json`:

```
{  
  "projectName": "AnytownDumpsters",  
  "placementName": "customer217",  
  "attributes": {
```

```
    "phone": "123-456-7890",  
    "location": "123 Any Street Anytown, USA 10001"  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [UpdatePlacement](#) na Referência de AWS CLI Comandos.

update-project

O código de exemplo a seguir mostra como usar update-project.

AWS CLI

Para atualizar as configurações de um projeto

O update-project exemplo a seguir atualiza a descrição de um projeto.

```
aws iot1click-projects update-project \  
  --project-name AnytownDumpsters \  
  --description "All dumpsters (yard waste, recycling, garbage) in the Anytown  
region."
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar o AWS IoT 1-Click com AWS CLI o no Guia do desenvolvedor do IoT AWS 1-Click](#).

- Para API obter detalhes, consulte [UpdateProject](#) na Referência de AWS CLI Comandos.

AWS IoT Analytics exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Analytics.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-put-message

O código de exemplo a seguir mostra como usar batch-put-message.

AWS CLI

Para enviar uma mensagem para um canal

O batch-put-message exemplo a seguir envia uma mensagem para o canal especificado.

```
aws iotanalytics batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://batch-put-message.json
```

Conteúdo de batch-put-message.json:

```
{  
  "channelName": "mychannel",  
  "messages": [  
    {  
      "messageId": "0001",  
      "payload": "eyJhdGVtcGVyYXR1cmUiOiAyMCB9"  
    }  
  ]  
}
```

Saída:

```
{  
  "batchPutMessageErrorEntries": []  
}
```

Para obter mais informações, consulte a [BatchPutMessage](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [BatchPutMessage](#) na Referência de AWS CLI Comandos.

cancel-pipeline-reprocessing

O código de exemplo a seguir mostra como usar `cancel-pipeline-reprocessing`.

AWS CLI

Para cancelar o reprocessamento de dados por meio de um pipeline

O `cancel-pipeline-reprocessing` exemplo a seguir cancela o reprocessamento de dados por meio do pipeline especificado.

```
aws iotanalytics cancel-pipeline-reprocessing \  
  --pipeline-name mypipeline \  
  --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

Este comando não produz saída.

Para obter mais informações, consulte a [CancelPipelineReprocessing](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [CancelPipelineReprocessing](#) na Referência de AWS CLI Comandos.

create-channel

O código de exemplo a seguir mostra como usar `create-channel`.

AWS CLI

Para criar um canal

O `create-channel` exemplo a seguir cria um canal com a configuração especificada. Um canal coleta dados de um MQTT tópico e arquiva as mensagens brutas e não processadas antes de publicar os dados em um pipeline.

```
aws iotanalytics create-channel \  
  --cli-input-json file://create-channel.json
```

Conteúdo de `create-channel.json`:

```
{
  "channelName": "mychannel",
  "retentionPeriod": {
    "unlimited": true
  },
  "tags": [
    {
      "key": "Environment",
      "value": "Production"
    }
  ]
}
```

Saída:

```
{
  "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",
  "channelName": "mychannel",
  "retentionPeriod": {
    "unlimited": true
  }
}
```

Para obter mais informações, consulte a [CreateChannel](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [CreateChannel](#) na Referência de AWS CLI Comandos.

create-dataset-content

O código de exemplo a seguir mostra como usar `create-dataset-content`.

AWS CLI

Para criar o conteúdo de um conjunto de dados

O `create-dataset-content` exemplo a seguir cria o conteúdo do conjunto de dados especificado aplicando uma `queryAction` (uma SQL consulta) ou uma `containerAction` (executando um aplicativo em contêiner).

```
aws iotanalytics create-dataset-content \
  --dataset-name mydataset
```

Saída:

```
{
  "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"
}
```

Para obter mais informações, consulte a [CreateDatasetContent](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [CreateDatasetContent](#) na Referência de AWS CLI Comandos.

create-dataset

O código de exemplo a seguir mostra como usar `create-dataset`.

AWS CLI

Para criar um conjunto de dados

O `create-dataset` exemplo a seguir cria um conjunto de dados. Um conjunto de dados armazena dados recuperados de um armazenamento de dados aplicando uma `queryAction` (uma SQL consulta) ou uma `containerAction` (executando um aplicativo em contêiner). Esta operação cria o esqueleto de um conjunto de dados. Você pode preencher o conjunto de dados manualmente ligando `CreateDatasetContent` ou automaticamente de acordo com a especificação que `trigger` você especificar.

```
aws iotanalytics create-dataset \
  --cli-input-json file://create-dataset.json
```

Conteúdo de `create-dataset.json`:

```
{
  "datasetName": "mydataset",
  "actions": [
    {
      "actionName": "myDatasetAction",
      "queryAction": {
        "sqlQuery": "SELECT * FROM mydatastore"
      }
    }
  ],
}
```

```
"retentionPeriod": {
  "unlimited": true
},
"tags": [
  {
    "key": "Environment",
    "value": "Production"
  }
]
}
```

Saída:

```
{
  "datasetName": "mydataset",
  "retentionPeriod": {
    "unlimited": true
  },
  "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

Para obter mais informações, consulte a [CreateDataset](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [CreateDataset](#) na Referência de AWS CLI Comandos.

create-datastore

O código de exemplo a seguir mostra como usar create-datastore.

AWS CLI

Para criar um armazenamento de dados

O create-datastore exemplo a seguir cria um armazenamento de dados, que é um repositório de mensagens.

```
aws iotanalytics create-datastore \
  --cli-input-json file://create-datastore.json
```

Conteúdo de create-datastore.json:

```
{
```

```
"datastoreName": "mydatastore",
"retentionPeriod": {
  "numberOfDays": 90
},
"tags": [
  {
    "key": "Environment",
    "value": "Production"
  }
]
}
```

Saída:

```
{
  "datastoreName": "mydatastore",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
mydatastore",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

Para obter mais informações, consulte a [CreateDatastore](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [CreateDatastore](#) na Referência de AWS CLI Comandos.

create-pipeline

O código de exemplo a seguir mostra como usar `create-pipeline`.

AWS CLI

Crie um pipeline de IoT Analytics

O `create-pipeline` exemplo a seguir cria um pipeline. Um pipeline consome mensagens de um canal e permite que você as processe antes de armazená-las em um datastore. Você deve especificar tanto um canal quanto uma atividade de armazenamento de dados e, opcionalmente, até 23 atividades adicionais na `pipelineActivities` matriz.

```
aws iotanalytics create-pipeline \
```

```
--cli-input-json file://create-pipeline.json
```

Conteúdo de 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"
    }
  ]
}
```

Saída:

```
{
  "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
  "pipelineName": "mypipeline"
}
```

```
}
```

Para obter mais informações, consulte a [CreatePipeline](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [CreatePipeline](#) na Referência de AWS CLI Comandos.

delete-channel

O código de exemplo a seguir mostra como usar `delete-channel`.

AWS CLI

Excluir um canal do IoT Analytics

O `delete-channel` exemplo a seguir exclui o canal especificado.

```
aws iotanalytics delete-channel \  
  --channel-name mychannel
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeleteChannel](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DeleteChannel](#) na Referência de AWS CLI Comandos.

delete-dataset-content

O código de exemplo a seguir mostra como usar `delete-dataset-content`.

AWS CLI

Para excluir o conteúdo do conjunto de dados

O `delete-dataset-content` exemplo a seguir exclui o conteúdo do conjunto de dados especificado.

```
aws iotanalytics delete-dataset-content \  
  --dataset-name mydataset
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeleteDatasetContent](#)Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DeleteDatasetContent](#)na Referência de AWS CLI Comandos.

delete-dataset

O código de exemplo a seguir mostra como usar delete-dataset.

AWS CLI

Para excluir um conjunto de dados

O delete-dataset exemplo a seguir exclui o conjunto de dados especificado. Você não precisa excluir o conteúdo do conjunto de dados antes de executar esta operação.

```
aws iotanalytics delete-dataset \  
  --dataset-name mydataset
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeleteDataset](#)Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DeleteDataset](#)na Referência de AWS CLI Comandos.

delete-datastore

O código de exemplo a seguir mostra como usar delete-datastore.

AWS CLI

Para excluir um armazenamento de dados

O delete-datastore exemplo a seguir exclui o armazenamento de dados especificado.

```
aws iotanalytics delete-datastore \  
  --datastore-name mydatastore
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeleteDatastore](#)Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DeleteDatastore](#)na Referência de AWS CLI Comandos.

delete-pipeline

O código de exemplo a seguir mostra como usar `delete-pipeline`.

AWS CLI

Para excluir um pipeline

O `delete-pipeline` exemplo a seguir exclui o pipeline especificado.

```
aws iotanalytics delete-pipeline \  
  --pipeline-name mypipeline
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeletePipeline](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DeletePipeline](#) na Referência de AWS CLI Comandos.

describe-channel

O código de exemplo a seguir mostra como usar `describe-channel`.

AWS CLI

Para recuperar informações sobre um canal

O `describe-channel` exemplo a seguir exibe detalhes, incluindo estatísticas, do canal especificado.

```
aws iotanalytics describe-channel \  
  --channel-name mychannel \  
  --include-statistics
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte a [DescribeChannel](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DescribeChannel](#) na Referência de AWS CLI Comandos.

describe-dataset

O código de exemplo a seguir mostra como usar describe-dataset.

AWS CLI

Para recuperar informações sobre um conjunto de dados

O describe-dataset exemplo a seguir exibe detalhes do conjunto de dados especificado.

```
aws iotanalytics describe-dataset \
  --dataset-name mydataset
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte a [DescribeDataset](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DescribeDataset](#) na Referência de AWS CLI Comandos.

describe-datastore

O código de exemplo a seguir mostra como usar `describe-datastore`.

AWS CLI

Para recuperar informações sobre um armazenamento de dados

O `describe-datastore` exemplo a seguir exibe detalhes, incluindo estatísticas, do armazenamento de dados especificado.

```

aws iotanalytics describe-datastore \
  --datastore-name mydatastore \
  --include-statistics

```

Saída:

```

{
  "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
    }
  }
}
```

Para obter mais informações, consulte a [DescribeDatastore](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DescribeDatastore](#) na Referência de AWS CLI Comandos.

describe-logging-options

O código de exemplo a seguir mostra como usar `describe-logging-options`.

AWS CLI

Para recuperar as opções de registro atuais

O `describe-logging-options` exemplo a seguir exibe as opções atuais de registro do AWS IoT Analytics.

```
aws iotanalytics describe-logging-options
```

Este comando não produz saída. Saída:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
    "enabled": true,
    "level": "ERROR"
  }
}
```

Para obter mais informações, consulte a [DescribeLoggingOptions](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DescribeLoggingOptions](#) na Referência de AWS CLI Comandos.

describe-pipeline

O código de exemplo a seguir mostra como usar `describe-pipeline`.

AWS CLI

Para recuperar informações sobre um pipeline

O `describe-pipeline` exemplo a seguir exibe detalhes do pipeline especificado.

```
aws iotanalytics describe-pipeline \  
  --pipeline-name mypipeline
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}  
}
```

Para obter mais informações, consulte a [DescribePipeline](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [DescribePipeline](#) na Referência de AWS CLI Comandos.

get-dataset-content

O código de exemplo a seguir mostra como usar `get-dataset-content`.

AWS CLI

Para recuperar o conteúdo de um conjunto de dados

O `get-dataset-content` exemplo a seguir recupera o conteúdo de um conjunto de dados conforme pré-assinado. URIs

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Saída:

```
{  
  "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=..."  
    }  
  ]  
}
```

Para obter mais informações, consulte [GetDatasetContent](#) guia.

- Para API obter detalhes, consulte [GetDatasetContent](#) na Referência de AWS CLI Comandos.

list-channels

O código de exemplo a seguir mostra como usar `list-channels`.

AWS CLI

Para recuperar uma lista de canais

O `list-channels` exemplo a seguir exibe informações resumidas dos canais disponíveis.

```
aws iotanalytics list-channels
```

Saída:

```
{
  "channelSummaries": [
    {
      "status": "ACTIVE",
      "channelName": "mychannel",
      "creationTime": 1557860351.001,
      "lastUpdateTime": 1557860351.001
    }
  ]
}
```

Para obter mais informações, consulte a [ListChannels](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [ListChannels](#) na Referência de AWS CLI Comandos.

list-dataset-contents

O código de exemplo a seguir mostra como usar `list-dataset-contents`.

AWS CLI

Para listar informações sobre o conteúdo do conjunto de dados

O `list-dataset-contents` exemplo a seguir lista informações sobre o conteúdo do conjunto de dados que foi criado.

```
aws iotanalytics list-dataset-contents \
  --dataset-name mydataset
```

Saída:

```
{
  "datasetContentSummaries": [
    {
      "status": {
        "state": "SUCCEEDED"
      },
      "scheduleTime": 1557863215.995,
      "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",
      "creationTime": 1557863215.995
    }
  ]
}
```

Para obter mais informações, consulte a [ListDatasetContents](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [ListDatasetContents](#) na Referência de AWS CLI Comandos.

list-datasets

O código de exemplo a seguir mostra como usar `list-datasets`.

AWS CLI

Para recuperar informações sobre conjuntos de dados

O `list-datasets` exemplo a seguir lista informações resumidas sobre os conjuntos de dados disponíveis.

```
aws iotanalytics list-datasets
```

Saída:

```
{
  "datasetSummaries": [
    {
      "status": "ACTIVE",
      "datasetName": "mydataset",
      "lastUpdateTime": 1557859240.658,
      "triggers": [],
    }
  ]
}
```



```
    "creationTime": 1557859240.658,
    "actions": [
      {
        "actionName": "query_32",
        "actionType": "QUERY"
      }
    ]
  }
]
```

Para obter mais informações, consulte a [ListDatasets](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [ListDatasets](#) na Referência de AWS CLI Comandos.

list-datastores

O código de exemplo a seguir mostra como usar `list-datastores`.

AWS CLI

Para recuperar uma lista de armazenamentos de dados

O `list-datastores` exemplo a seguir exibe informações resumidas sobre os armazenamentos de dados disponíveis.

```
aws iotanalytics list-datastores
```

Saída:

```
{
  "datastoreSummaries": [
    {
      "status": "ACTIVE",
      "datastoreName": "mydatastore",
      "creationTime": 1557858971.02,
      "lastUpdateTime": 1557858971.02
    }
  ]
}
```

Para obter mais informações, consulte a [ListDatastores](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [ListDatastores](#) na Referência de AWS CLI Comandos.

list-pipelines

O código de exemplo a seguir mostra como usar `list-pipelines`.

AWS CLI

Para recuperar uma lista de pipelines

O `list-pipelines` exemplo a seguir exibe uma lista dos pipelines disponíveis.

```
aws iotanalytics list-pipelines
```

Saída:

```
{
  "pipelineSummaries": [
    {
      "pipelineName": "mypipeline",
      "creationTime": 1557859124.432,
      "lastUpdateTime": 1557859124.432,
      "reprocessingSummaries": []
    }
  ]
}
```

Para obter mais informações, consulte a [ListPipelines](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [ListPipelines](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags que você anexou ao recurso especificado.

```
aws iotanalytics list-tags-for-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

Saída:

```
{  
  "tags": [  
    {  
      "value": "bar",  
      "key": "foo"  
    }  
  ]  
}
```

Para obter mais informações, consulte a [ListTagsForResource](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-logging-options

O código de exemplo a seguir mostra como usar `put-logging-options`.

AWS CLI

Para definir ou atualizar as opções de registro

O `put-logging-options` exemplo a seguir define ou atualiza as opções de registro do AWS IoT Analytics. Se você atualizar o valor de qualquer `loggingOptions` campo, pode levar até um minuto para que a alteração entre em vigor. Além disso, se você alterar a política anexada à função especificada no campo `roleArn` (por exemplo, para corrigir uma política inválida), pode levar até cinco minutos para que a alteração entre em vigor.

```
aws iotanalytics put-logging-options \  
  --cli-input-json file://put-logging-options.json
```

Conteúdo de `put-logging-options.json`:

```
{  
  "loggingOptions": {
```

```
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
    "level": "ERROR",
    "enabled": true
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte a [PutLoggingOptions](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [PutLoggingOptions](#) na Referência de AWS CLI Comandos.

run-pipeline-activity

O código de exemplo a seguir mostra como usar run-pipeline-activity.

AWS CLI

Para simular uma atividade de pipeline

O run-pipeline-activity exemplo a seguir simula os resultados da execução de uma atividade de pipeline em uma carga de mensagem.

```
aws iotanalytics run-pipeline-activity \
  --pipeline-activity file://maths.json \
  --payloads file://payloads.json
```

Conteúdo de maths.json:

```
{
  "math": {
    "name": "MyMathActivity",
    "math": "((temp - 32) * 5.0) / 9.0",
    "attribute": "tempC"
  }
}
```

Conteúdo de payloads.json:

```
[
```

```

    {"humidity": 52, "temp": 68 },
    {"humidity": 52, "temp": 32 }
  ]

```

Saída:

```

{
  "logResult": "",
  "payloads": [
    "eyJodW1pZG10eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
    "eyJodW1pZG10eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
  ]
}

```

Para obter mais informações, consulte a [RunPipelineActivity](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [RunPipelineActivity](#) na Referência de AWS CLI Comandos.

sample-channel-data

O código de exemplo a seguir mostra como usar `sample-channel-data`.

AWS CLI

Para recuperar mensagens de amostra de um canal

O `sample-channel-data` exemplo a seguir recupera uma amostra de mensagens do canal especificado ingeridas durante o período especificado. Você pode recuperar até 10 mensagens.

```

aws iotanalytics sample-channel-data \
  --channel-name mychannel

```

Saída:

```

{
  "payloads": [
    "eyJhdGVtcGVyYXR1cmUiOiAyMjB9",
    "eyJhZm9vIjogImJhcnVzIj0="
  ]
}

```

Para obter mais informações, consulte a [SampleChannelData](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [SampleChannelData](#) na Referência de AWS CLI Comandos.

start-pipeline-reprocessing

O código de exemplo a seguir mostra como usar `start-pipeline-reprocessing`.

AWS CLI

Para iniciar o reprocessamento do pipeline

O `start-pipeline-reprocessing` exemplo a seguir inicia o reprocessamento de dados brutos de mensagens por meio do pipeline especificado.

```
aws iotanalytics start-pipeline-reprocessing \  
  --pipeline-name mypipeline
```

Saída:

```
{  
  "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"  
}
```

Para obter mais informações, consulte a [StartPipelineReprocessing](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [StartPipelineReprocessing](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar ou modificar tags para um recurso

O `tag-resource` exemplo a seguir adiciona ou modifica as tags anexadas ao recurso especificado.

```
aws iotanalytics tag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tags "[{"key": "Environment", "value": "Production"}]"
```

Este comando não produz saída.

Para obter mais informações, consulte a [TagResource](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Como remover as tags de um recurso

O `untag-resource` exemplo a seguir remove as tags com os nomes de chave especificados do recurso especificado.

```
aws iotanalytics untag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tag-keys ["Environment"]
```

Este comando não produz saída.

Para obter mais informações, consulte `UntagResource < https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html >` na Referência do IoT AWS Analytics. API

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-channel

O código de exemplo a seguir mostra como usar `update-channel`.

AWS CLI

Para modificar um canal

O `update-channel` exemplo a seguir modifica as configurações do canal especificado.

```
aws iotanalytics update-channel \  
  --cli-input-json file://update-channel.json
```

Conteúdo de `update-channel.json`:

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte a [UpdateChannel](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [UpdateChannel](#) na Referência de AWS CLI Comandos.

update-dataset

O código de exemplo a seguir mostra como usar `update-dataset`.

AWS CLI

Para atualizar um conjunto de dados

O `update-dataset` exemplo a seguir modifica as configurações do conjunto de dados especificado.

```
aws iotanalytics update-dataset \  
  --cli-input-json file://update-dataset.json
```

Conteúdo de `update-dataset.json`:

```
{  
  "datasetName": "mydataset",  
  "actions": [  
    {  
      "actionName": "myDatasetUpdateAction",  
      "queryAction": {  
        "sqlQuery": "SELECT * FROM mydatastore"  
      }  
    }  
  ]  
}
```



```
    }
  ],
  "retentionPeriod": {
    "numberOfDays": 92
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte [UpdateDataset](https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html) < https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html > na Referência do IoT AWS Analytics. API

- Para API obter detalhes, consulte [UpdateDataset](#) na Referência de AWS CLI Comandos.

update-datastore

O código de exemplo a seguir mostra como usar `update-datastore`.

AWS CLI

Para atualizar um armazenamento de dados

O `update-datastore` exemplo a seguir modifica as configurações do armazenamento de dados especificado.

```
aws iotanalytics update-datastore \
  --cli-input-json file://update-datastore.json
```

Conteúdo do `update-datastore.json`:

```
{
  "datastoreName": "mydatastore",
  "retentionPeriod": {
    "numberOfDays": 93
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte a [UpdateDatastore](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [UpdateDatastore](#) na Referência de AWS CLI Comandos.

update-pipeline

O código de exemplo a seguir mostra como usar update-pipeline.

AWS CLI

Para atualizar um pipeline

O update-pipeline exemplo a seguir modifica as configurações do pipeline especificado. Você deve especificar um canal e uma atividade de armazenamento de dados e, opcionalmente, até 23 atividades adicionais, na pipelineActivities matriz.

```
aws iotanalytics update-pipeline \  
  --cli-input-json file://update-pipeline.json
```

Conteúdo do 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"  
      }  
    }  
  ]  
}
```

```
}
```

Este comando não produz saída.

Para obter mais informações, consulte a [UpdatePipeline](#) Referência do AWS IoT API Analytics.

- Para API obter detalhes, consulte [UpdatePipeline](#) na Referência de AWS CLI Comandos.

Exemplos do Device Advisor usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Device Advisor.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-suite-definition

O código de exemplo a seguir mostra como usar `create-suite-definition`.

AWS CLI

Exemplo 1: Para criar uma suíte de testes do IoT Device Advisor

O `create-suite-definition` exemplo a seguir cria uma suíte de testes do Device Advisor na AWS IoT com a configuração de definição de suíte especificada.

```
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"
  
```

Saída:

```

{
  "suiteDefinitionId": "0jtsgio7yenu",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/0jtsgio7yenu",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
  
```

Para obter mais informações, consulte [Criar uma definição de suíte de testes](#) no Guia do desenvolvedor do AWS IoT Core.

Exemplo 2: Para criar um pacote de testes de qualificação mais recente do IoT Device Advisor

O `create-suite-definition` exemplo a seguir cria uma suíte de testes de qualificação do Device Advisor com a versão mais recente na AWS IoT com a configuração de definição de suíte especificada.

```

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"
  }'
  
```

Saída:

```

{
  "suiteDefinitionId": "txgsuolk2myj",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/txgsuolk2myj",
}
  
```

```
"suiteDefinitionName": "TestSuiteName",  
"createdAt": "2022-12-02T11:38:13.263000-05:00"  
}
```

Para obter mais informações, consulte [Criar uma definição de suíte de testes](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [CreateSuiteDefinition](#) na Referência de AWS CLI Comandos.

delete-suite-definition

O código de exemplo a seguir mostra como usar `delete-suite-definition`.

AWS CLI

Para excluir o conjunto de testes do IoT Device Advisor

O `delete-suite-definition` exemplo a seguir exclui a suíte de testes do Device Advisor com a ID de definição da suíte especificada.

```
aws iotdeviceadvisor delete-suite-definition \  
--suite-definition-id 0jtsgio7yenu
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteSuiteDefinition](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [DeleteSuiteDefinition](#) na Referência de AWS CLI Comandos.

get-endpoint

O código de exemplo a seguir mostra como usar `get-endpoint`.

AWS CLI

Exemplo 1: Para obter as informações sobre um endpoint no nível da conta do IoT Device Advisor

O `get-endpoint` exemplo a seguir obtém as informações sobre um endpoint de teste em nível de conta do Device Advisor.

```
aws iotdeviceadvisor get-endpoint
```

Saída:

```
{
  "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Exemplo 2: Para obter as informações sobre um endpoint em nível de dispositivo do IoT Device Advisor

O `get-endpoint` exemplo a seguir obtém as informações sobre um endpoint de teste em nível de dispositivo do Device Advisor com o `thing-arn` ou `certificate-arn` especificado.

```
aws iotdeviceadvisor get-endpoint \
  --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIoTThing
```

Saída:

```
{
  "endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Para obter mais informações, consulte [Obtenha um endpoint de teste](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [GetEndpoint](#) na Referência de AWS CLI Comandos.

get-suite-definition

O código de exemplo a seguir mostra como usar `get-suite-definition`.

AWS CLI

Para obter as informações sobre uma suíte de testes do IoT Device Advisor

O `get-suite-definition` exemplo a seguir obtém as informações sobre uma suíte de testes do Aevice Advisor com a ID de definição da suíte especificada.

```
aws iotdeviceadvisor get-suite-definition \
  --suite-definition-id qqcsmtyyjabl
```

Saída:

```
{
  "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": {}
}
```

Para obter mais informações, consulte [Obtenha uma definição de suíte de testes](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [GetSuiteDefinition](#) na Referência de AWS CLI Comandos.

get-suite-run-report

O código de exemplo a seguir mostra como usar `get-suite-run-report`.

AWS CLI

Para obter as informações sobre uma suíte de testes de qualificação do IoT Device Advisor, execute o relatório

O `get-suite-run-report` exemplo a seguir obtém o link de download do relatório para uma execução bem-sucedida da suíte de testes de qualificação do Device Advisor com a ID de definição da suíte e a ID de execução da suíte especificadas.

```
aws iotdeviceadvisor get-suite-run-report \
```

```
--suite-definition-id ztvb5aek4w4x \  
--suite-run-id p6awv83nre6v
```

Saída:

```
{  
  "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-  
prod.s3.amazonaws.com/report.downloadlink"  
}
```

Para obter mais informações, consulte [Obter um relatório de qualificação para uma suíte de testes de qualificação bem-sucedida executada](#) no Guia do Desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [GetSuiteRunReport](#) na Referência de AWS CLI Comandos.

get-suite-run

O código de exemplo a seguir mostra como usar `get-suite-run`.

AWS CLI

Para obter as informações sobre o status de execução de uma suíte de testes do IoT Device Advisor

O `get-suite-run` exemplo a seguir obtém as informações sobre o status de execução de uma suíte de testes do Device Advisor com a ID de definição da suíte e a ID de execução da suíte especificadas.

```
aws iotdeviceadvisor get-suite-run \  
--suite-definition-id qqcsmtyyjabl \  
--suite-run-id nzlfyhaa18oa
```

Saída:

```
{  
  "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": "uta5d9j1kvwc",
        "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": {}
}

```

Para [obter mais informações, consulte Execute uma suíte de testes](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [GetSuiteRun](#) na Referência de AWS CLI Comandos.

list-suite-definitions

O código de exemplo a seguir mostra como usar `list-suite-definitions`.

AWS CLI

Exemplo 1: Para listar os conjuntos de testes do IoT Device Advisor que você criou

O `list-suite-definitions` exemplo a seguir lista até 25 suítes de testes do Device Advisor que você criou na AWS IoT. Se você tiver mais de 25 suítes de testes, o `nextToken` "" será mostrado na saída. Você pode usar esse "nextToken" para mostrar o resto das suítes de teste que você criou.

```
aws iotdeviceadvisor list-suite-definitions
```

Saída:

```
{
  "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"
}
```

Exemplo 2: Para listar os conjuntos de testes do IoT Device Advisor que você criou com as configurações especificadas

O `list-suite-definitions` exemplo a seguir lista os conjuntos de testes do Device Advisor que você criou na AWS IoT com o número máximo de resultados especificado. Se você tiver mais suítes de teste do que o número máximo, o `nextToken` "" será mostrado na saída. Se você tiver

"nextToken", você pode usar "nextToken" para mostrar as suítes de teste que você criou e que não foram mostradas antes.

```
aws iotdeviceadvisor list-suite-definitions \
  --max-result 1 \
  --next-token "nextTokenValue"
```

Saída:

```
{
  "suiteDefinitionInformationList": [
    {
      "suiteDefinitionId": "ztlvb5aew4w4x",
      "suiteDefinitionName": "TestSuite2",
      "defaultDevices": [],
      "intendedForQualification": true,
      "isLongDurationTest": false,
      "protocol": "MqttV3_1_1",
      "createdAt": "2022-11-17T14:15:56.830000-05:00"
    }
  ],
  "nextToken": "nextTokenValue"
}
```

Para obter mais informações, consulte [ListSuiteDefinitions](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListSuiteDefinitions](#) na Referência de AWS CLI Comandos.

list-suite-runs

O código de exemplo a seguir mostra como usar `list-suite-runs`.

AWS CLI

Exemplo 1: Para listar todas as informações sobre o status de execução do pacote de testes do IoT Device Advisor especificado

O `list-suite-runs` exemplo a seguir lista todas as informações sobre o status de execução de uma suíte de testes do Device Advisor com a ID de definição da suíte especificada. Se você tiver mais de 25 execuções de suítes de testes, o `nextToken` "" será mostrado na saída. Você pode usar esse "nextToken" para mostrar o resto das execuções da suíte de testes.

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id ztlvb5aew4w4x
```

Saída:

```
{  
  "suiteRunsList": [  
    {  
      "suiteDefinitionId": "ztlvb5aew4w4x",  
      "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  
    }  
  ]  
}
```

Exemplo 2: Para listar informações sobre o status de execução do pacote de testes do IoT Device Advisor especificado com as configurações especificadas

O `list-suite-runs` exemplo a seguir lista informações sobre o status de execução de uma suíte de testes do Device Advisor com a ID de definição da suíte especificada e o número máximo de resultados especificado. Se você tiver mais execuções de suítes de testes do que o número máximo, o `nextToken` "" será mostrado na saída. Se você tiver "nextToken", você pode usar "nextToken" para mostrar as execuções da suíte de testes que não foram mostradas antes.

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id qqcsmtyyjaml \  
  --max-result 1 \  
  --next-token "nextTokenValue"
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [ListSuiteRuns](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [ListSuiteRuns](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags anexadas a um recurso do IoT Device Advisor

O `list-tags-for-resource` exemplo a seguir lista as tags anexadas a um recurso do Device Advisor. O recurso do consultor de dispositivos pode ser um `Suitedefinition-Arn` ou um `Suiterun-Arn`.

```

aws iotdeviceadvisor list-tags-for-resource \
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny

```

Saída:

```

{
  "tags": {
    "TestTagKey": "TestTagValue"
  }
}

```

```
}
```

Para obter mais informações, consulte [ListTagsForResource](#) a API Referência de AWS IoT e os [tipos de recursos definidos pelo AWS IoT Core Device Advisor](#) na Referência de Autorização de Serviço.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

start-suite-run

O código de exemplo a seguir mostra como usar `start-suite-run`.

AWS CLI

Para iniciar uma suíte de testes do IoT Device Advisor, execute

O `start-suite-run` exemplo a seguir lista os widgets disponíveis na sua AWS conta.

```
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"}}'
```

Saída:

```
{
  "suiteRunId": "pwmucgw7lt9s",
  "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/qqcsmtyyjabl/pwmucgw7lk9s",
  "createdAt": "2022-12-02T15:43:05.581000-05:00"
}
```

Para obter mais informações, consulte [Iniciar a execução de uma suíte de testes](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [StartSuiteRun](#) na Referência de AWS CLI Comandos.

stop-suite-run

O código de exemplo a seguir mostra como usar `stop-suite-run`.

AWS CLI

Para interromper uma suíte de testes do IoT Device Advisor que está em execução no momento

O `stop-suite-run` exemplo a seguir interrompe uma suíte de testes do Device Advisor que está sendo executada atualmente com a ID de definição da suíte e a ID de execução da suíte especificadas.

```
aws iotdeviceadvisor stop-suite-run \  
  --suite-definition-id qqcsmtyyjabl \  
  --suite-run-id nzlfyh18oa
```

Este comando não produz saída.

Para obter mais informações, consulte [Interromper a execução de um conjunto de testes](#) no Guia do desenvolvedor do AWS IoT Core.

- Para API obter detalhes, consulte [StopSuiteRun](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar e modificar as tags existentes de um recurso do IoT Device Advisor

O `tag-resource` exemplo a seguir adiciona e modifica as tags existentes de um recurso do Device Advisor com o arn e as tags do recurso especificados. O recurso do consultor de dispositivos pode ser um `SuiteDefinition-Arn` ou um `SuiteRun-Arn`.

```
aws iotdeviceadvisor tag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tags '{"TagKey": "TagValue"}
```

Este comando não produz saída.

Para obter mais informações, consulte [TagResource](#) a API Referência de AWS IoT e os [tipos de recursos definidos pelo AWS IoT Core Device Advisor](#) na Referência de Autorização de Serviço.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover as tags existentes de um recurso do IoT Device Advisor

O `untag-resource` exemplo a seguir remove as tags existentes de um recurso do Device Advisor com o recurso `arn` e a chave de tag especificados. O recurso do consultor de dispositivos pode ser um `SuiteDefinition-Arn` ou um `Suiterun-Arn`.

```
aws iotdeviceadvisor untag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tag-keys "TagKey"
```

Este comando não produz saída.

Para obter mais informações, consulte [UntagResource](#) a API Referência de AWS IoT e os [tipos de recursos definidos pelo AWS IoT Core Device Advisor](#) na Referência de Autorização de Serviço.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-suite-definition

O código de exemplo a seguir mostra como usar `update-suite-definition`.

AWS CLI

Exemplo 1: Para atualizar um conjunto de testes do IoT Device Advisor

O `update-suite-definition` exemplo a seguir atualiza uma suíte de testes do Device Advisor na AWS IoT com o ID de definição de suíte e a configuração de definição de suíte especificados.

```
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"
  }

```

Saída:

```

{
  "suiteDefinitionId": "3hsn88h4p2g5",
  "suiteDefinitionName": "TestSuiteName",
  "suiteDefinitionVersion": "v3",
  "createdAt": "2022-11-17T14:15:56.830000-05:00",
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}

```

Exemplo 2: Para atualizar um conjunto de testes de qualificação do IoT Device Advisor

O `update-suite-definition` exemplo a seguir atualiza uma suíte de testes de qualificação do Device Advisor na AWS IoT com o ID de definição de suíte e a configuração de definição de suíte especificados.

```

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"}'

```

Saída:

```

{
  "suiteDefinitionId": "txgsuolk2myj",
  "suiteDefinitionName": "TestSuiteName",
  "suiteDefinitionVersion": "v3",
  "createdAt": "2022-11-17T14:15:56.830000-05:00",
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}

```

Para obter mais informações, consulte [UpdateSuiteDefinition](#) na Referência de AWS IoT. API

- Para API obter detalhes, consulte [UpdateSuiteDefinition](#) na Referência de AWS CLI Comandos.

AWS IoT data exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT data.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-thing-shadow

O código de exemplo a seguir mostra como usar `delete-thing-shadow`.

AWS CLI

Para excluir o documento paralelo de um dispositivo

O `delete-thing-shadow` exemplo a seguir exclui todo o documento de sombra do dispositivo chamado `MyRPi`.

```
aws iot-data delete-thing-shadow \  
  --thing-name MyRPi \  
  "output.txt"
```

O comando não produz saída na tela, mas `output.txt` contém informações que confirmam a versão e a data e hora do documento paralelo que você excluiu.

```
{"version":2,"timestamp":1560270384}
```

Para obter mais informações, consulte [Usando sombras no Guia](#) para desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [DeleteThingShadow](#) na Referência de AWS CLI Comandos.

get-thing-shadow

O código de exemplo a seguir mostra como usar `get-thing-shadow`.

AWS CLI

Para obter uma coisa, documento de sombra

O `get-thing-shadow` exemplo a seguir obtém o documento sombra da coisa para a coisa de IoT especificada.

```
aws iot-data get-thing-shadow \  
  --thing-name MyRPI \  
  output.txt
```

O comando não produz nenhuma saída na tela, mas o seguinte mostra o conteúdo de `output.txt`:

```
{  
  "state":{  
    "reported":{  
      "moisture":"low"  
    }  
  },  
  "metadata":{  
    "reported":{  
      "moisture":{  
        "timestamp":1560269319  
      }  
    }  
  },  
  "version":1,"timestamp":1560269405  
}
```

Para obter mais informações, consulte [Fluxo de dados do Device Shadow Service](#) no AWS IoT Developers Guide.

- Para API obter detalhes, consulte [GetThingShadow](#) na Referência de AWS CLI Comandos.

update-thing-shadow

O código de exemplo a seguir mostra como usar `update-thing-shadow`.

AWS CLI

Para atualizar uma sombra de coisa

O `update-thing-shadow` exemplo a seguir modifica o estado atual da sombra do dispositivo para a coisa especificada e a salva no arquivo `output.txt`.

```
aws iot-data update-thing-shadow \  
  --thing-name MyRPi \  
  --payload '{"state":{"reported":{"moisture":"okay"}}}' \  
  output.txt
```

O comando não produz nenhuma saída na tela, mas o seguinte mostra o conteúdo de `output.txt`:

```
{  
  "state": {  
    "reported": {  
      "moisture": "okay"  
    }  
  },  
  "metadata": {  
    "reported": {  
      "moisture": {  
        "timestamp": 1560270036  
      }  
    }  
  },  
  "version": 2,  
  "timestamp": 1560270036  
}
```

Para obter mais informações, consulte [Fluxo de dados do Device Shadow Service](#) no AWS IoT Developers Guide.

- Para API obter detalhes, consulte [UpdateThingShadow](#) na Referência de AWS CLI Comandos.

AWS IoT Events exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Events.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-put-message

O código de exemplo a seguir mostra como usar `batch-put-message`.

AWS CLI

Para enviar mensagens (entradas) para AWS IoT Events

O `batch-put-message` exemplo a seguir envia um conjunto de mensagens para o sistema AWS IoT Events. Cada carga útil de mensagem é transformada na entrada que você especifica (`inputName`) e ingerida em qualquer detector que monitore essa entrada. Se várias mensagens forem enviadas, a ordem em que as mensagens são processadas não é garantida. Para garantir o pedido, você deve enviar mensagens uma de cada vez e aguardar uma resposta bem-sucedida.

```
aws iotevents-data batch-put-message \  
  --cli-input-json file://highPressureMessage.json
```

Conteúdo de `highPressureMessage.json`:

```
{  
  "messages": [  
    {  
      "inputName": "highPressureMessage"  
    }  
  ]  
}
```

```

    {
      "messageId": "00001",
      "inputName": "PressureInput",
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
80, \"temperature\": 39} }"
    }
  ]
}

```

Saída:

```

{
  "BatchPutMessageErrorEntries": []
}

```

Para obter mais informações, consulte a [BatchPutMessage](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [BatchPutMessage](#) na Referência de AWS CLI Comandos.

batch-update-detector

O código de exemplo a seguir mostra como usar batch-update-detector.

AWS CLI

Para atualizar um detector (instância)

O batch-update-detector exemplo a seguir atualiza o estado, os valores das variáveis e as configurações do cronômetro de um ou mais detectores (instâncias) de um modelo de detector especificado.

```

aws iotevents-data batch-update-detector \
  --cli-input-json file://budFulton-A32.json

```

Conteúdo de budFulton-A32.json:

```

{
  "detectors": [
    {
      "messageId": "00001",

```

```

    "detectorModelName": "motorDetectorModel",
    "keyValue": "Fulton-A32",
    "state": {
      "stateName": "Normal",
      "variables": [
        {
          "name": "pressureThresholdBreached",
          "value": "0"
        }
      ],
      "timers": [
      ]
    }
  ]
}

```

Saída:

```

{
  "batchUpdateDetectorErrorEntries": []
}

```

Para obter mais informações, consulte a [BatchUpdateDetector](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [BatchUpdateDetector](#) na Referência de AWS CLI Comandos.

create-detector-model

O código de exemplo a seguir mostra como usar `create-detector-model`.

AWS CLI

Para criar um modelo de detector

O `create-detector-model` exemplo a seguir cria um modelo de detector com sua configuração especificada por um arquivo de parâmetros.

```

aws iotevents create-detector-model \
  --cli-input-json file://motorDetectorModel.json

```

Conteúdo de 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": "$variable.pressureThresholdBreached >
1",
            "actions": [
              {
                "sns": {
                  "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                }
              }
            ]
          }
        ],
      },
      "onInput": {
        "events": [
          {
            "eventName": "Overpressurized",
            "condition": "$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"

```



```
    "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"
  }
}
```

Para obter mais informações, consulte a [CreateDetectorModel](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [CreateDetectorModel](#) na Referência de AWS CLI Comandos.

create-input

O código de exemplo a seguir mostra como usar create-input.

AWS CLI

Para criar uma entrada

O create-input exemplo a seguir cria uma entrada.

```
aws iotevents create-input \
  --cli-input-json file://pressureInput.json
```

Conteúdo de pressureInput.json:

```
{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte a [CreateInput](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [CreateInput](#) na Referência de AWS CLI Comandos.

delete-detector-model

O código de exemplo a seguir mostra como usar `delete-detector-model`.

AWS CLI

Para excluir um modelo de detector

O `delete-detector-model` exemplo a seguir exclui o modelo de detector especificado. Todas as instâncias ativas do modelo do detector também são excluídas.

```
aws iotevents delete-detector-model \
  --detector-model-name motorDetectorModel
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeleteDetectorModel](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [DeleteDetectorModel](#) na Referência de AWS CLI Comandos.

delete-input

O código de exemplo a seguir mostra como usar `delete-input`.

AWS CLI

Para excluir uma entrada

O `delete-input` exemplo a seguir exclui a entrada especificada.

```
aws iotevents delete-input \  
  --input-name PressureInput
```

Este comando não produz saída.

Para obter mais informações, consulte a [DeleteInput](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [DeleteInput](#) na Referência de AWS CLI Comandos.

`describe-detector-model`

O código de exemplo a seguir mostra como usar `describe-detector-model`.

AWS CLI

Para obter informações sobre um modelo de detector

O `describe-detector-model` exemplo a seguir exibe detalhes do modelo de detector especificado. Como o `version` parâmetro não foi especificado, as informações sobre a versão mais recente são retornadas.

```
aws iotevents describe-detector-model \  
  --detector-model-name motorDetectorModel
```

Saída:

```
{  
  "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":
"pressureThresholdBreach",
                    "value":
"$variable.pressureThresholdBreach + 3"
                  }
                ],
                "condition":
"$input.PressureInput.sensorData.pressure > 70",
                "nextState": "Dangerous"
              }
            ],
            "events": []
          },
          "stateName": "Normal",
          "onEnter": {
            "events": [
              {
                "eventName": "init",
                "actions": [
                  {
                    "setVariable": {
                      "variableName":
"pressureThresholdBreach",
                      "value": "0"
                    }
                  ]
                },
                "condition": "true"
              }
            ]
          }
        ]
      },
    ],
  },
}
```

```

        "onExit": {
            "events": []
        }
    },
    {
        "onInput": {
            "transitionEvents": [
                {
                    "eventName": "BackToNormal",
                    "actions": [],
                    "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreach <= 1",
                    "nextState": "Normal"
                }
            ],
            "events": [
                {
                    "eventName": "Overpressurized",
                    "actions": [
                        {
                            "setVariable": {
                                "variableName":
"pressureThresholdBreach",
                                "value": "3"
                            }
                        }
                    ],
                    "condition":
"$input.PressureInput.sensorData.pressure > 70"
                },
                {
                    "eventName": "Pressure Okay",
                    "actions": [
                        {
                            "setVariable": {
                                "variableName":
"pressureThresholdBreach",
                                "value":
"$variable.pressureThresholdBreach - 1"
                            }
                        }
                    ]
                }
            ]
        }
    }
}

```

```

        "condition":
"$input.PressureInput.sensorData.pressure <= 70"
        }
    ]
},
"stateName": "Dangerous",
"onEnter": {
    "events": [
        {
            "eventName": "Pressure Threshold Breached",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                    }
                }
            ],
            "condition": "$variable.pressureThresholdBreached >
1"
        }
    ]
},
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                    }
                }
            ],
            "condition": "true"
        }
    ]
}
},
"initialStateName": "Normal"
}
}

```



```
}
```

Para obter mais informações, consulte a [DescribeDetectorModel](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [DescribeDetectorModel](#) na Referência de AWS CLI Comandos.

describe-detector

O código de exemplo a seguir mostra como usar describe-detector.

AWS CLI

Para obter informações sobre um detector (instância).

O describe-detector exemplo a seguir exibe detalhes do detector especificado (instância).

```
aws iotevents-data describe-detector \  
  --detector-model-name motorDetectorModel \  
  --key-value "Fulton-A32"
```

Saída:

```
{  
  "detector": {  
    "lastUpdateTime": 1560797852.776,  
    "creationTime": 1560797852.775,  
    "state": {  
      "variables": [  
        {  
          "name": "pressureThresholdBreach",  
          "value": "3"  
        }  
      ],  
      "stateName": "Dangerous",  
      "timers": []  
    },  
    "keyValue": "Fulton-A32",  
    "detectorModelName": "motorDetectorModel",  
    "detectorModelVersion": "1"  
  }  
}
```

```
}
```

Para obter mais informações, consulte a [DescribeDetector](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [DescribeDetector](#) na Referência de AWS CLI Comandos.

describe-input

O código de exemplo a seguir mostra como usar `describe-input`.

AWS CLI

Para obter informações sobre uma entrada

O `describe-input` exemplo a seguir exibe detalhes da entrada especificada.

```
aws iotevents describe-input \  
  --input-name PressureInput
```

Saída:

```
{  
  "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"  
        }  
      ]  
    }  
  }  
}
```

```
    }  
  }  
}
```

Para obter mais informações, consulte a [DescribeInput](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [DescribeInput](#) na Referência de AWS CLI Comandos.

describe-logging-options

O código de exemplo a seguir mostra como usar `describe-logging-options`.

AWS CLI

Para obter informações sobre as configurações de registro

O `describe-logging-options` exemplo a seguir recupera as configurações atuais das opções de registro do AWS IoT Events.

```
aws iotevents describe-logging-options
```

Saída:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "enabled": false,  
    "level": "ERROR"  
  }  
}
```

Para obter mais informações, consulte a [DescribeLoggingOptions](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [DescribeLoggingOptions](#) na Referência de AWS CLI Comandos.

list-detector-model-versions

O código de exemplo a seguir mostra como usar `list-detector-model-versions`.

AWS CLI

Para obter informações sobre as versões de um modelo de detector

O `list-detector-model-versions` exemplo a seguir lista todas as versões de um modelo de detector. Somente os metadados associados a cada versão de modelo de detector são retornados.

```
aws iotevents list-detector-model-versions \  
  --detector-model-name motorDetectorModel
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte a [ListDetectorModelVersions](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [ListDetectorModelVersions](#) na Referência de AWS CLI Comandos.

list-detector-models

O código de exemplo a seguir mostra como usar `list-detector-models`.

AWS CLI

Para obter uma lista de seus modelos de detectores

O `list-detector-models` exemplo a seguir lista os modelos de detectores que você criou. Somente os metadados associados a cada modelo de detector são retornados.

```
aws iotevents list-detector-models
```

Saída:

```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

Para obter mais informações, consulte a [ListDetectorModels](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [ListDetectorModels](#) na Referência de AWS CLI Comandos.

list-detectors

O código de exemplo a seguir mostra como usar `list-detectors`.

AWS CLI

Para obter uma lista de detectores para um modelo de detector

O `list-detectors` exemplo a seguir lista os detectores (as instâncias de um modelo de detector) em sua conta.

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

Saída:

```
{
  "detectorSummaries": [
    {
      "lastUpdateTime": 1558129925.2,
```

```
    "creationTime": 1552073155.527,
    "state": {
      "stateName": "Normal"
    },
    "keyValue": "Fulton-A32",
    "detectorModelName": "motorDetectorModel",
    "detectorModelVersion": "1"
  }
]
}
```

Para obter mais informações, consulte a [ListDetectors](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [ListDetectors](#) na Referência de AWS CLI Comandos.

list-inputs

O código de exemplo a seguir mostra como usar `list-inputs`.

AWS CLI

Para listar entradas

O `list-inputs` exemplo a seguir lista as entradas que você criou em sua conta.

```
aws iotevents list-inputs
```

Este comando não produz saída. Saída:

```
{
  {
    "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"
  }
}
```

Para obter mais informações, consulte a [ListInputs](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [ListInputs](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags atribuídas a um recurso.

O `list-tags-for-resource` exemplo a seguir lista os nomes e valores das chaves de tag que você atribuiu ao recurso.

```
aws iotevents list-tags-for-resource \  
  --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Saída:

```
{  
  "tags": [  
    {  
      "value": "motor",  
      "key": "deviceType"  
    }  
  ]  
}
```

Para obter mais informações, consulte a [ListTagsForResource](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-logging-options

O código de exemplo a seguir mostra como usar `put-logging-options`.

AWS CLI

Para definir opções de registro

O `put-logging-options` exemplo a seguir define ou atualiza as opções de registro de eventos do AWS IoT. Se você atualizar o valor de qualquer `loggingOptions` field, it can take up to one minute for the change to take effect. Also, if you change the policy attached to the role you specified in the ``roleArn` campo (por

exemplo, para corrigir uma política inválida), pode levar até cinco minutos para que a alteração entre em vigor.

```
aws iotevents put-logging-options \  
  --cli-input-json file://logging-options.json
```

Conteúdo de `logging-options.json`:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",  
    "enabled": true,  
    "detectorDebugOptions": [  
      {  
        "detectorModelName": "motorDetectorModel",  
        "keyValue": "Fulton-A32"  
      }  
    ]  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte a [PutLoggingOptions](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [PutLoggingOptions](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um recurso

O `tag-resource` exemplo a seguir adiciona ou modifica (se a chave `deviceType` já existir) a tag anexada ao recurso especificado.

```
aws iotevents tag-resource \  
  --cli-input-json file://pressureInput.tag.json
```


Conteúdo de `pressureInput.tag.json`:

```
{
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "tags": [
    {
      "key": "deviceType",
      "value": "motor"
    }
  ]
}
```

Este comando não produz saída.

Para obter mais informações, consulte a [TagResource](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Como remover as tags de um recurso

O `untag-resource` exemplo a seguir remove a tag com o nome de chave especificado do recurso especificado.

```
aws iotevents untag-resource \
  --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \
  --tagkeys deviceType
```

Este comando não produz saída.

Para obter mais informações, consulte a [UntagResource](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-detector-model

O código de exemplo a seguir mostra como usar `update-detector-model`.

AWS CLI

Para atualizar um modelo de detector

O `update-detector-model` exemplo a seguir atualiza o modelo de detector especificado. Os detectores (instâncias) gerados pela versão anterior são excluídos e recriados à medida que novas entradas chegam.

```
aws iotevents update-detector-model \  
  --cli-input-json file://motorDetectorModel.update.json
```

Conteúdo de `motorDetectorModel.update.json`:

```
{  
  "detectorModelName": "motorDetectorModel",  
  "detectorModelDefinition": {  
    "states": [  
      {  
        "stateName": "Normal",  
        "onEnter": {  
          "events": [  
            {  
              "eventName": "init",  
              "condition": "true",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreached",  
                    "value": "0"  
                  }  
                }  
              ]  
            }  
          ]  
        },  
        "onInput": {  
          "transitionEvents": [  
            {  
              "eventName": "Overpressurized",  
              "condition": "$input.PressureInput.sensorData.pressure >  
70",  
              "actions": [  
                {
```

```

        "setVariable": {
            "variableName": "pressureThresholdBreached",
            "value":
"$variable.pressureThresholdBreached + 3"
        }
    ],
    "nextState": "Dangerous"
}
]
}
},
{
    "stateName": "Dangerous",
    "onEnter": {
        "events": [
            {
                "eventName": "Pressure Threshold Breached",
                "condition": "$variable.pressureThresholdBreached > 1",
                "actions": [
                    {
                        "sns": {
                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                        }
                    }
                ]
            }
        ]
    },
    "onInput": {
        "events": [
            {
                "eventName": "Overpressurized",
                "condition": "$input.PressureInput.sensorData.pressure >
70",
                "actions": [
                    {
                        "setVariable": {
                            "variableName": "pressureThresholdBreached",
                            "value": "3"
                        }
                    }
                ]
            }
        ]
    }
}
]

```

```

        },
        {
            "eventName": "Pressure Okay",
            "condition": "$input.PressureInput.sensorData.pressure
<= 70",
            "actions": [
                {
                    "setVariable": {
                        "variableName": "pressureThresholdBreached",
                        "value":
"$variable.pressureThresholdBreached - 1"
                    }
                }
            ]
        }
    ],
    "transitionEvents": [
        {
            "eventName": "BackToNormal",
            "condition": "$input.PressureInput.sensorData.pressure
<= 70 && $variable.pressureThresholdBreached <= 1",
            "nextState": "Normal"
        }
    ]
},
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "condition": "true",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                    }
                }
            ]
        }
    ]
}
},
"initialStateName": "Normal"

```

```

    },
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
  }

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte a [UpdateDetectorModel](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [UpdateDetectorModel](#) na Referência de AWS CLI Comandos.

update-input

O código de exemplo a seguir mostra como usar `update-input`.

AWS CLI

Para atualizar uma entrada

O `update-input` exemplo a seguir atualiza a entrada especificada com uma nova descrição e definição.

```

aws iotevents update-input \
  --cli-input-json file://pressureInput.json

```

Conteúdo de `pressureInput.json`:

```

{

```

```
"inputName": "PressureInput",
"inputDescription": "Pressure readings from a motor",
"inputDefinition": {
  "attributes": [
    { "jsonPath": "sensorData.pressure" },
    { "jsonPath": "motorid" }
  ]
}
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte a [UpdateInput](#) Referência de API Eventos de AWS IoT.

- Para API obter detalhes, consulte [UpdateInput](#) na Referência de AWS CLI Comandos.

AWS IoT Events-Data exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Events-Data.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-put-message

O código de exemplo a seguir mostra como usar batch-put-message.

AWS CLI

Para enviar mensagens (entradas) para AWS IoT Events

O batch-put-message exemplo a seguir envia um conjunto de mensagens para o sistema AWS IoT Events. Cada carga útil de mensagem é transformada na entrada que você especifica (inputName) e ingerida em qualquer detector que monitore essa entrada. Se várias mensagens forem enviadas, a ordem em que as mensagens são processadas não é garantida. Para garantir o pedido, você deve enviar mensagens uma de cada vez e aguardar uma resposta bem-sucedida.

```
aws iotevents-data batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://highPressureMessage.json
```

Conteúdo de highPressureMessage.json:

```
{  
  "messages": [  
    {  
      "messageId": "00001",  
      "inputName": "PressureInput",  
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":  
80, \"temperature\": 39} }"  
    }  
  ]  
}
```

Saída:

```
{  
  "BatchPutMessageErrorEntries": []  
}
```

Para obter mais informações, consulte [BatchPutMessage](#) no [AWS IoT Events Developer Guide*](#).

- Para API obter detalhes, consulte [BatchPutMessage](#) na Referência de AWS CLI Comandos.

batch-update-detector

O código de exemplo a seguir mostra como usar batch-update-detector.

AWS CLI

Para atualizar um detector (instância)

O batch-update-detector exemplo a seguir atualiza o estado, os valores das variáveis e as configurações do cronômetro de um ou mais detectores (instâncias) de um modelo de detector especificado.

```
aws iotevents-data batch-update-detector \  
  --cli-input-json file://budFulton-A32.json
```

Conteúdo de budFulton-A32.json:

```
{  
  "detectors": [  
    {  
      "messageId": "00001",  
      "detectorModelName": "motorDetectorModel",  
      "keyValue": "Fulton-A32",  
      "state": {  
        "stateName": "Normal",  
        "variables": [  
          {  
            "name": "pressureThresholdBreach",  
            "value": "0"  
          }  
        ],  
        "timers": [  
        ]  
      }  
    }  
  ]  
}
```

Saída:


```
{
  "batchUpdateDetectorErrorEntries": []
}
```

Para obter mais informações, consulte [BatchUpdateDetector](#) no *AWS IoT Events Developer Guide**

- Para API obter detalhes, consulte [BatchUpdateDetector](#) na Referência de AWS CLI Comandos.

create-detector-model

O código de exemplo a seguir mostra como usar `create-detector-model`.

AWS CLI

Para criar um modelo de detector

O `create-detector-model` exemplo a seguir cria um modelo de detector.

```
aws iotevents create-detector-model \
  --cli-input-json file://motorDetectorModel.json
```

Conteúdo de `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": "$input.PressureInput.sensorData.pressure
&gt; 70",
        "actions": [
            {
                "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value": "3"
                }
            }
        ]
    },
    {
        "eventName": "Pressure Okay",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70",
        "actions": [
            {
                "setVariable": {
                    "variableName": "pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached - 1"
                }
            }
        ]
    }
],
"transitionEvents": [
    {
        "eventName": "BackToNormal",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 &amp;&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"
}

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [CreateDetectorModel](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [CreateDetectorModel](#) na Referência de AWS CLI Comandos.

create-input

O código de exemplo a seguir mostra como usar create-input.

AWS CLI

Para criar uma entrada

O `create-input` exemplo a seguir cria uma entrada.

```
aws iotevents create-input \  
  --cli-input-json file://pressureInput.json
```

Conteúdo de `pressureInput.json`:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [CreateInput](#) em *AWS IoT Events Developer Guide**

- Para API obter detalhes, consulte [CreateInput](#) na Referência de AWS CLI Comandos.

delete-detector-model

O código de exemplo a seguir mostra como usar `delete-detector-model`.

AWS CLI

Para excluir um modelo de detector

O `delete-detector-model` exemplo a seguir exclui um modelo de detector. Todas as instâncias ativas do modelo do detector também são excluídas.

```
aws iotevents delete-detector-model \  
  --detector-model-name motorDetectorModel*
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteDetectorModel](#) no *AWS IoT Events Developer Guide**.

- Para API obter detalhes, consulte [DeleteDetectorModel](#) na Referência de AWS CLI Comandos.

delete-input

O código de exemplo a seguir mostra como usar `delete-input`.

AWS CLI

Para excluir uma entrada

O `delete-input` exemplo a seguir exclui uma entrada.

```
aws iotevents delete-input \  
  --input-name PressureInput
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteInput](#) no *AWS IoT Events Developer Guide**.

- Para API obter detalhes, consulte [DeleteInput](#) na Referência de AWS CLI Comandos.

describe-detector-model

O código de exemplo a seguir mostra como usar `describe-detector-model`.

AWS CLI

Para obter informações sobre um modelo de detector

O `describe-detector-model` exemplo a seguir descreve um modelo de detector. Se o `version` parâmetro não for especificado, o comando retornará informações sobre a versão mais recente.

```
aws iotevents describe-detector-model \  
--detector-model-name motorDetectorModel
```

Saída:

```
{  
  "detectorModel": {  
    "detectorModelConfiguration": {  
      "status": "ACTIVE",  
      "lastUpdateTime": 1560796816.077,  
      "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
      "creationTime": 1560796816.077,  
      "detectorModelArn": "arn:aws:iotevents:us-  
west-2:123456789012:detectorModel/motorDetectorModel",  
      "key": "motorid",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    },  
    "detectorModelDefinition": {  
      "states": [  
        {  
          "onInput": {  
            "transitionEvents": [  
              {  
                "eventName": "Overpressurized",  
                "actions": [  
                  {  
                    "setVariable": {  
                      "variableName":  
"pressureThresholdBreached",  
                      "value":  
"$variable.pressureThresholdBreached + 3"  
                    }  
                  }  
                ],  
                "condition":  
"$input.PressureInput.sensorData.pressure > 70",  
                "nextState": "Dangerous"  
              }  
            ],  
            "events": []  
          },  
          "stateName": "Normal",  
        }  
      ]  
    }  
  }  
}
```

```

        "onEnter": {
            "events": [
                {
                    "eventName": "init",
                    "actions": [
                        {
                            "setVariable": {
                                "variableName":
"pressureThresholdBreach",
                                "value": "0"
                            }
                        }
                    ],
                    "condition": "true"
                }
            ]
        },
        "onExit": {
            "events": []
        }
    },
    {
        "onInput": {
            "transitionEvents": [
                {
                    "eventName": "BackToNormal",
                    "actions": [],
                    "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreach <= 1",
                    "nextState": "Normal"
                }
            ],
            "events": [
                {
                    "eventName": "Overpressurized",
                    "actions": [
                        {
                            "setVariable": {
                                "variableName":
"pressureThresholdBreach",
                                "value": "3"
                            }
                        }
                    ]
                }
            ]
        }
    }
}

```



```

        ],
        "condition":
"$input.PressureInput.sensorData.pressure > 70"
    },
    {
        "eventName": "Pressure Okay",
        "actions": [
            {
                "setVariable": {
                    "variableName":
"pressureThresholdBreached",
                    "value":
"$variable.pressureThresholdBreached - 1"
                }
            }
        ],
        "condition":
"$input.PressureInput.sensorData.pressure <= 70"
    }
]
},
"stateName": "Dangerous",
"onEnter": {
    "events": [
        {
            "eventName": "Pressure Threshold Breached",
            "actions": [
                {
                    "sns": {
                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                    }
                }
            ]
        },
        {
            "condition": "$variable.pressureThresholdBreached >
1"
        }
    ]
},
"onExit": {
    "events": [
        {
            "eventName": "Normal Pressure Restored",
            "actions": [

```



```
"lastUpdateTime": 1560797852.776,  
"creationTime": 1560797852.775,  
"state": {  
  "variables": [  
    {  
      "name": "pressureThresholdBreached",  
      "value": "3"  
    }  
  ],  
  "stateName": "Dangerous",  
  "timers": []  
},  
"keyValue": "Fulton-A32",  
"detectorModelName": "motorDetectorModel",  
"detectorModelVersion": "1"  
}  
}
```

Para obter mais informações, consulte [DescribeDetector](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [DescribeDetector](#) na Referência de AWS CLI Comandos.

describe-input

O código de exemplo a seguir mostra como usar `describe-input`.

AWS CLI

Para obter informações sobre uma entrada

O `describe-input` exemplo a seguir recupera os detalhes de uma entrada.

```
aws iotevents describe-input \  
  --input-name PressureInput
```

Saída:

```
{  
  "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"
            }
        ]
    }
}
}
}

```

Para obter mais informações, consulte [Descreva o Input](#) em *AWS IoT Events Developer Guide**.

- Para API obter detalhes, consulte [Descreva o Input](#) na Referência de AWS CLI Comandos.

describe-logging-options

O código de exemplo a seguir mostra como usar `describe-logging-options`.

AWS CLI

Para obter informações sobre as configurações de registro

O `describe-logging-options` exemplo a seguir recupera as opções atuais de registro do AWS IoT Events.

```
aws iotevents describe-logging-options
```

Saída:

```

{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "enabled": false,
    "level": "ERROR"
  }
}

```

```
}
```

Para obter mais informações, consulte [DescribeLoggingOptions](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [DescribeLoggingOptions](#) na Referência de AWS CLI Comandos.

list-detector-model-versions

O código de exemplo a seguir mostra como usar `list-detector-model-versions`.

AWS CLI

Para obter informações sobre as versões de um modelo de detector

O `list-detector-model-versions` exemplo a seguir lista todas as versões de um modelo de detector. Somente os metadados associados a cada versão de modelo de detector são retornados.

```
aws iotevents list-detector-model-versions \  
  --detector-model-name motorDetectorModel
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListDetectorModelVersions](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [ListDetectorModelVersions](#) na Referência de AWS CLI Comandos.

list-detector-models

O código de exemplo a seguir mostra como usar `list-detector-models`.

AWS CLI

Para obter uma lista de seus modelos de detectores

O `list-detector-models` exemplo a seguir lista os modelos de detectores que você criou. Somente os metadados associados a cada modelo de detector são retornados.

```
aws iotevents list-detector-models
```

Saída:

```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

Para obter mais informações, consulte [ListDetectorModels](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [ListDetectorModels](#) na Referência de AWS CLI Comandos.

list-detectors

O código de exemplo a seguir mostra como usar `list-detectors`.

AWS CLI

Para obter uma lista de detectores para um modelo de detector

O `list-detectors` exemplo a seguir lista os detectores (as instâncias de um modelo de detector).

```
aws iotevents-data list-detectors \  
  --detector-model-name motorDetectorModel
```

Saída:

```
{  
  "detectorSummaries": [  
    {  
      "lastUpdateTime": 1558129925.2,  
      "creationTime": 1552073155.527,  
      "state": {  
        "stateName": "Normal"  
      },  
      "keyValue": "Fulton-A32",  
      "detectorModelName": "motorDetectorModel",  
      "detectorModelVersion": "1"  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListDetectors](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [ListDetectors](#) na Referência de AWS CLI Comandos.

list-inputs

O código de exemplo a seguir mostra como usar `list-inputs`.

AWS CLI

Para listar as entradas

O `list-inputs` exemplo a seguir lista as entradas que você criou.

```
aws iotevents list-inputs
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [ListInputs](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [ListInputs](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags atribuídas a um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags (metadados) que você atribuiu ao recurso.

```
aws iotevents list-tags-for-resource \  
--resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Saída:

```
{  
  "tags": [  
    {  
      "value": "motor",  
      "key": "deviceType"  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListTagsForResource](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-logging-options

O código de exemplo a seguir mostra como usar `put-logging-options`.

AWS CLI

Para definir as opções de registro

O `list-tags-for-resource` exemplo a seguir define ou atualiza as opções de registro de eventos do AWS IoT. Se você atualizar o valor de qualquer campo `loggingOptions`, levará até um minuto para ver a mudança entrar em vigor. Além disso, se você alterar a política anexada à função especificada no `roleArn` campo (por exemplo, para corrigir uma política inválida), levará até cinco minutos para que a alteração entre em vigor.

```
aws iotevents put-logging-options \  
  --cli-input-json file://logging-options.json
```

Conteúdo de `logging-options.json`:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",  
    "enabled": true,  
    "detectorDebugOptions": [  
      {  
        "detectorModelName": "motorDetectorModel",  
        "keyValue": "Fulton-A32"  
      }  
    ]  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [PutLoggingOptions](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [PutLoggingOptions](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um recurso

O `tag-resource` exemplo a seguir adiciona ou modifica as tags de um determinado recurso. As tags são metadados que podem ser usados para gerenciar um recurso.

```
aws iotevents tag-resource \  
  --cli-input-json file://pressureInput.tag.json
```

Conteúdo de `pressureInput.tag.json`:

```
{  
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
  "tags": [  
    {  
      "key": "deviceType",  
      "value": "motor"  
    }  
  ]  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [TagResource](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Como remover as tags de um recurso

O `untag-resource` exemplo a seguir remove as tags especificadas do recurso.

```
aws iotevents untag-resource \  
  --cli-input-json file://pressureInput.untag.json
```

Conteúdo de `pressureInput.untag.json`:

```
{  
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
  "tagKeys": [  
    "deviceType"  
  ]  
}
```

```

        "deviceType"
    ]
}

```

Este comando não produz saída.

Para obter mais informações, consulte [UntagResource](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-detector-model

O código de exemplo a seguir mostra como usar `update-detector-model`.

AWS CLI

Para atualizar um modelo de detector

O `update-detector-model` exemplo a seguir atualiza um modelo de detector. Os detectores (instâncias) gerados pela versão anterior são excluídos e recriados à medida que novas entradas chegam.

```

aws iotevents update-detector-model \
  --cli-input-json file://motorDetectorModel.update.json

```

Conteúdo de `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": "$input.PressureInput.sensorData.pressure > 70",
  "actions": [
    {
      "setVariable": {
        "variableName": "pressureThresholdBreached",
        "value": "3"
      }
    }
  ]
},
{
  "eventName": "Pressure Okay",
  "condition": "$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"
}

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [UpdateDetectorModel](#) no AWS IoT Events Developer Guide*.

- Para API obter detalhes, consulte [UpdateDetectorModel](#) na Referência de AWS CLI Comandos.

update-input

O código de exemplo a seguir mostra como usar `update-input`.

AWS CLI

Para atualizar uma entrada

O `update-input` exemplo a seguir atualiza uma entrada.

```
aws iotevents update-input \  
--cli-input-json file://pressureInput.json
```

Conteúdo de `pressureInput.json`:

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [UpdateInput](#) no *AWS IoT Events Developer Guide**

- Para API obter detalhes, consulte [UpdateInput](#) na Referência de AWS CLI Comandos.

AWS IoT Greengrass exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Greengrass.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-role-to-group

O código de exemplo a seguir mostra como usar `associate-role-to-group`.

AWS CLI

Para associar uma função a um grupo do Greengrass

O `associate-role-to-group` exemplo a seguir associa a IAM função especificada a um grupo do Greengrass. A função de grupo é usada por funções e conectores locais do Lambda para acessar serviços. AWS Por exemplo, seu papel de grupo pode conceder as permissões necessárias para a integração do CloudWatch Logs.

```
aws greengrass associate-role-to-group \
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \
  --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

Saída:

```
{
  "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

Para obter mais informações, consulte [Configurar a função do grupo](#) no Guia do AWS desenvolvedor do IoT Greengrass.

- Para API obter detalhes, consulte [AssociateRoleToGroup](#) na Referência de AWS CLI Comandos.

associate-service-role-to-account

O código de exemplo a seguir mostra como usar `associate-service-role-to-account`.

AWS CLI

Para associar uma função de serviço à sua AWS conta

O `associate-service-role-to-account` exemplo a seguir associa uma função IAM de serviço, especificada por `itsARN`, ao AWS IoT AWS Greengrass em sua conta. Você deve ter criado previamente a função de serviço em IAM e deve associar a ela um documento de política que permita que o AWS IoT Greengrass assuma essa função.

```
aws greengrass associate-service-role-to-account \  
  --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

Saída:

```
{  
  "AssociatedAt": "2019-06-25T18:12:45Z"  
}
```

Para obter mais informações, consulte [Greengrass Service Role](#) no IoT AWS Greengrass Developer Guide.

- Para API obter detalhes, consulte [AssociateServiceRoleToAccount](#) na Referência de AWS CLI Comandos.

`create-connector-definition-version`

O código de exemplo a seguir mostra como usar `create-connector-definition-version`.

AWS CLI

Para criar uma versão de definição de conector

O `create-connector-definition-version` exemplo a seguir cria uma versão de definição de conector e a associa à definição de conector especificada. Todos os conectores em uma versão definem valores para seus parâmetros.

```
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:greenrass-TwilioAuthToken-Slp6\\", \\"TwilioAuthTokenSecretArn-ResourceId\\": \\"TwilioAuthToken\\", \\"DefaultFromPhoneNumber\\": \\"4254492999\\"}]"]"
```

Saída:

```
{
  "Arn": "arn:aws:greenrass:us-west-2:123456789012:/greenrass/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"
}
```

- Para API obter detalhes, consulte [CreateConnectorDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-connector-definition

O código de exemplo a seguir mostra como usar `create-connector-definition`.

AWS CLI

Para criar uma definição de conector

O `create-connector-definition` exemplo a seguir cria uma definição de conector e uma versão inicial de definição de conector. A versão inicial contém um conector. Todos os conectores em uma versão definem valores para seus parâmetros.

```
aws greengrass create-connector-definition \
  --name MySNSConnector \
  --initial-version "{\\"Connectors\\": [{"\\"Id\\":\\"MySNSConnector\\",\\"ConnectorArn\\":\\"arn:aws:greenrass:us-west-2:/connectors/SNS/versions/1\\",\\"Parameters\\": {"\\"DefaultSNSArn\\":\\"arn:aws:sns:us-west-2:123456789012:GGConnectorTopic\\"}"}]"}"
```

Saída:

```
{
  "Arn": "arn:aws:greenrass:us-west-2:123456789012:/greenrass/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"
}
```

Para obter mais informações, consulte [Introdução aos Greengrass Connectors \(CLI\)](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateConnectorDefinition](#) na Referência de AWS CLI Comandos.

create-core-definition-version

O código de exemplo a seguir mostra como usar `create-core-definition-version`.

AWS CLI

Para criar uma versão de definição principal

O `create-core-definition-version` exemplo a seguir cria uma versão de definição principal e a associa à definição de núcleo especificada. A versão pode conter apenas um núcleo. Antes de criar um núcleo, você deve primeiro criar e provisionar a coisa de AWS IoT correspondente. Esse processo inclui os `iot` comandos a seguir, que retornam o `ThingArn` e o `CertificateArn` necessário para o `create-core-definition-version` comando.

Crie a coisa de AWS IoT que corresponde ao dispositivo principal:

```
aws iot create-thing \
  --thing-name "MyCoreDevice"
```

Saída:

```
{
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
  "thingName": "MyCoreDevice",
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

Crie chaves públicas e privadas e o certificado do dispositivo principal para a coisa. Este exemplo usa o `create-keys-and-certificate` comando e exige permissões de gravação no diretório atual. Como alternativa, você pode usar o comando `create-certificate-from-csr`.

```
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"
```

Saída:

```
{
  "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"
}
```

Crie uma política de AWS IoT que permita `iot` e `greengrass` aja. Para simplificar, a política a seguir permite ações em todos os recursos, mas sua política deve ser mais restritiva.

```
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
\": [\"*\"]}]}\""
```

Saída:

```
{
  "policyName": "Core_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
  "policyDocument": "{\n\"Version\":\n\"2012-10-17\", \"Statement\": [{\n\"Effect\n\n\":\n\"Allow\", \"Action\": [\n\"iot:Publish\", \n\"iot:Subscribe\", \n\"iot:Connect\n\n\", \n\"iot:Receive\"], \"Resource\": [\n\"*\n\n\"]}, {\n\"Effect\":\n\"Allow\", \"Action\":\n\n[\n\"iot:GetThingShadow\", \n\"iot:UpdateThingShadow\", \n\"iot>DeleteThingShadow\"],\n\n\"Resource\": [\n\"*\n\n\"]}, {\n\"Effect\":\n\"Allow\", \"Action\": [\n\"greengrass:*\"], \"Resource\n\n\": [\n\"*\n\n\"]}]}",
  "policyVersionId": "1"
}
```

Anexe a política ao certificado:

```
aws iot attach-policy \
  --policy-name "Core_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Este comando não produz saída.

Anexe o item ao certificado:

```
aws iot attach-thing-principal \
  --thing-name "MyCoreDevice" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Este comando não produz saída.

Crie a versão da definição principal:

```
aws greengrass create-core-definition-version \
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
  --cores "[{\n\"Id\":\n\"MyCoreDevice\", \"ThingArn\":\n\n\"arn:aws:iot:us-
west-2:123456789012:thing/MyCoreDevice\", \"CertificateArn\":\n\n\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz\n\n\", \n\n\"SyncShadow\":true}]"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Configurar o AWS IoT Greengrass Core no Guia do Desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [CreateCoreDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-core-definition

O código de exemplo a seguir mostra como usar `create-core-definition`.

AWS CLI

Exemplo 1: Para criar uma definição de núcleo vazia

O `create-core-definition` exemplo a seguir cria uma definição básica vazia (sem versão inicial) do Greengrass. Antes que o núcleo seja utilizável, você deve usar o `create-core-definition-version` comando para fornecer os outros parâmetros para o núcleo.

```
aws greengrass create-core-definition \
  --name cliGroup_Core
```

Saída:

```
{
  "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"
}
```

```
}

```

Exemplo 2: Para criar uma definição principal com uma versão inicial

O `create-core-definition` exemplo a seguir cria uma definição principal que contém uma versão inicial da definição principal. A versão pode conter apenas um núcleo. Antes de criar um núcleo, você deve primeiro criar e provisionar a coisa de AWS IoT correspondente. Esse processo inclui os `iot` comandos a seguir, que retornam o `ThingArn` e o `CertificateArn` necessário para o `create-core-definition` comando.

Crie a coisa de AWS IoT que corresponde ao dispositivo principal:

```
aws iot create-thing \
  --thing-name "MyCoreDevice"

```

Saída:

```
{
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
  "thingName": "MyCoreDevice",
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

Crie chaves públicas e privadas e o certificado do dispositivo principal para a coisa. Este exemplo usa o `create-keys-and-certificate` comando e exige permissões de gravação no diretório atual. Como alternativa, você pode usar o comando `create-certificate-from-csr`.

```
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"

```

Saída:

```
{
  "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-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
      "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIABAKCAQEAAqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcN0K4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
    },
    "certificateId":
    "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
  }

```

Crie uma política de AWS IoT que permita `iot` e `greengrass` aja. Para simplificar, a política a seguir permite ações em todos os recursos, mas sua política deve ser mais restritiva.

```

aws iot create-policy \
  --policy-name "Core_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\n:\n\"Allow\n\", \"Action\":[\"iot:Publish\n\", \"iot:Subscribe\n\", \"iot:Connect
\n\", \"iot:Receive\n\"], \"Resource\":[\"*\n\"]}, {\"Effect\":[\"Allow\n\", \"Action\":[
\n\"iot:GetThingShadow\n\", \"iot:UpdateThingShadow\n\", \"iot>DeleteThingShadow\n\"],
\n\"Resource\":[\"*\n\"]}, {\"Effect\":[\"Allow\n\", \"Action\":[\"greengrass:*\"], \"Resource
\n:[\"*\n\"]}]]}"

```

Saída:

```

{
  "policyName": "Core_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
  "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\n:\n\"Allow\n\", \"Action\":[\"iot:Publish\n\", \"iot:Subscribe\n\", \"iot:Connect
\n\", \"iot:Receive\n\"], \"Resource\":[\"*\n\"]}, {\"Effect\":[\"Allow\n\", \"Action\":[
\n\"iot:GetThingShadow\n\", \"iot:UpdateThingShadow\n\", \"iot>DeleteThingShadow\n\"],
\n\"Resource\":[\"*\n\"]}, {\"Effect\":[\"Allow\n\", \"Action\":[\"greengrass:*\"], \"Resource
\n:[\"*\n\"]}]]}",
  "policyVersionId": "1"
}

```

Anexe a política ao certificado:

```

aws iot attach-policy \
  --policy-name "Core_Devices" \

```



```
--target "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Este comando não produz saída.

Anexe o item ao certificado:

```
aws iot attach-thing-principal \
  --thing-name "MyCoreDevice" \
  --principal "arn:aws:iot:us-west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Este comando não produz saída.

Crie a definição principal:

```
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 } ] }"
```

Saída:

```
{
  "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/cc87b5b3-8f4b-465d-944c-1d6de5dbfcd",
  "Name": "MyCores",
  "LastUpdatedTimestamp": "2019-09-18T00:11:06.197Z",
  "LatestVersion": "cc87b5b3-8f4b-465d-944c-1d6de5dbfcd",
  "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"
}
```

Para obter mais informações, consulte [Configurar o AWS IoT Greengrass Core no Guia do Desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [CreateCoreDefinition](#) na Referência de AWS CLI Comandos.

create-deployment

O código de exemplo a seguir mostra como usar `create-deployment`.

AWS CLI

Para criar uma implantação para uma versão de um grupo do Greengrass

O `create-deployment` exemplo a seguir implanta a versão especificada de um grupo do Greengrass.

```
aws greengrass create-deployment \  
  --deployment-type NewDeployment \  
  --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \  
  --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Introdução aos conectores \(CLI\)](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateDeployment](#) na Referência de AWS CLI Comandos.

create-device-definition-version

O código de exemplo a seguir mostra como usar `create-device-definition-version`.

AWS CLI

Para criar uma versão de definição de dispositivo

O `create-device-definition-version` exemplo a seguir cria uma versão de definição de dispositivo e a associa à definição de dispositivo especificada. A versão define dois dispositivos. Antes de criar um dispositivo Greengrass, você deve primeiro criar e provisionar a coisa de IoT

correspondente AWS . Esse processo inclui os seguintes `iot` comandos que você deve executar para obter as informações necessárias para o comando `Greengrass`:

Crie a coisa de AWS IoT que corresponde ao dispositivo:

```
aws iot create-thing \
  --thing-name "InteriorTherm"
```

Saída:

```
{
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
  "thingName": "InteriorTherm",
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

Crie chaves públicas e privadas e o certificado do dispositivo para a coisa. Este exemplo usa o `create-keys-and-certificate` comando e exige permissões de gravação no diretório atual. Como alternativa, você pode usar o `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"
```

Saída:

```
{
  "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-----\nMIIEowIABAKCAQEAAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDCnOK4rMCxDR...fvY4+te\n-----END RSA PRIVATE KEY-----\n"
  }
}
```

```

    },
    "certificateId":
    "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}

```

Crie uma política de AWS IoT que permita `iot` e `greengrass` aja. Para simplificar, a política a seguir permite ações em todos os recursos, mas sua política pode ser mais restritiva:

```

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\":[\"*\"]}]}"

```

Saída:

```

{
  "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"
}

```

Anexe a política ao certificado:

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Anexe a coisa ao certificado

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \

```

```
--principal "arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Depois de criar e configurar a IoT conforme mostrado acima, use o ThingArn e CertificateArn dos dois primeiros comandos no exemplo a seguir.

```
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}]"
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [CreateDeviceDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-device-definition

O código de exemplo a seguir mostra como usar create-device-definition.

AWS CLI

Para criar uma definição de dispositivo

O create-device-definition exemplo a seguir cria uma definição de dispositivo que contém uma versão inicial da definição de dispositivo. A versão inicial define dois dispositivos. Antes de criar um dispositivo Greengrass, você deve primeiro criar e provisionar a coisa de IoT

correspondente AWS . Esse processo inclui os seguintes `iot` comandos que você deve executar para obter as informações necessárias para o comando `Greengrass`:

Crie a coisa de AWS IoT que corresponde ao dispositivo:

```
aws iot create-thing \
  --thing-name "InteriorTherm"
```

Saída:

```
{
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
  "thingName": "InteriorTherm",
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

Crie chaves públicas e privadas e o certificado do dispositivo para a coisa. Este exemplo usa o `create-keys-and-certificate` comando e exige permissões de gravação no diretório atual. Como alternativa, você pode usar o `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"
```

Saída:

```
{
  "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-----\nMIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjbmXDCnOK4rMCxDR...fvY4+te\n-----END RSA PRIVATE KEY-----\n"
  }
}
```

```

    },
    "certificateId":
    "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}

```

Crie uma política de AWS IoT que permita `iot` e `greengrass` aja. Para simplificar, a política a seguir permite ações em todos os recursos, mas sua política pode ser mais restritiva:

```

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\": [\"*\"]}]}\"

```

Saída:

```

{
  "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"
}

```

Anexe a política ao certificado:

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Anexe a coisa ao certificado

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \

```

```
--principal "arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

Depois de criar e configurar a IoT conforme mostrado acima, use o ThingArn e CertificateArn dos dois primeiros comandos no exemplo a seguir.

```
aws greengrass create-device-definition \
  --name "Sensors" \
  --initial-version "{ \"Devices\": [{ \"Id\": \"InteriorTherm\", \"ThingArn\": \"arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm\", \"CertificateArn\": \"arn:aws:iot:us-west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92\", \"SyncShadow\": true }, { \"Id\": \"ExteriorTherm\", \"ThingArn\": \"arn:aws:iot:us-west-2:123456789012:thing/ExteriorTherm\", \"CertificateArn\": \"arn:aws:iot:us-west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\", \"SyncShadow\": true } ] }
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [CreateDeviceDefinition](#) na Referência de AWS CLI Comandos.

create-function-definition-version

O código de exemplo a seguir mostra como usar create-function-definition-version.

AWS CLI

Para criar uma versão da definição da função

O `create-function-definition-version` exemplo a seguir cria uma nova versão da definição de função especificada. Essa versão especifica uma única função cujo ID é `Hello-World-function`, permite acesso ao sistema de arquivos e especifica um tamanho máximo de memória e um período de tempo limite.

```
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}}]}'
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [CreateFunctionDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-function-definition

O código de exemplo a seguir mostra como usar `create-function-definition`.

AWS CLI

Para criar uma definição de função Lambda

O `create-function-definition` exemplo a seguir cria uma definição de função Lambda e uma versão inicial fornecendo uma lista de funções Lambda (nesse caso, uma lista de apenas uma função chamada `TempMonitorFunction`) e suas configurações. Antes de criar a definição da função, você precisa da função Lambda. ARN Para criar a função e seu alias, use comandos `create-function` e `publish-version` comandos do Lambda. O `create-function`

comando do Lambda exige a função ARN de execução, embora o AWS IoT Greengrass não use essa função porque as permissões são especificadas na função de grupo do Greengrass. Você pode usar o IAM `create-role` comando para criar uma função vazia para usar com ARN a do Lambda `create-function` ou pode usar uma função de execução existente.

```
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}}]}\"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Como configurar o acesso a recursos locais usando a interface de linha de AWS comando](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateFunctionDefinition](#) na Referência de AWS CLI Comandos.

create-group-certificate-authority

O código de exemplo a seguir mostra como usar `create-group-certificate-authority`.

AWS CLI

Para criar uma autoridade de certificação (CA) para um grupo

O `create-group-certificate-authority` exemplo a seguir cria ou alterna uma CA para o grupo especificado.

```
aws greengrass create-group-certificate-authority \  
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
```

Saída:

```
{  
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/  
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/  
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEecc5"  
}
```

Para obter mais informações, consulte Segurança [AWS do IoT Greengrass no](#) Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateGroupCertificateAuthority](#) na Referência de AWS CLI Comandos.

create-group-version

O código de exemplo a seguir mostra como usar `create-group-version`.

AWS CLI

Para criar uma versão de um grupo do Greengrass

O `create-group-version` exemplo a seguir cria uma versão de grupo e a associa ao grupo especificado. A versão faz referência às versões principal, de recurso, de conector, de função e de assinatura que contêm as entidades a serem incluídas nessa versão de grupo. Você deve criar essas entidades antes de criar a versão do grupo.

Para criar uma definição de recurso com uma versão inicial, use o `create-resource-definition` comando. Para criar uma definição de conector com uma versão inicial, use o `create-connector-definition` comando. Para criar uma definição de função com uma versão inicial, use o `create-function-definition` comando. Para criar uma definição de assinatura com uma versão inicial, use o `create-subscription-definition` comando. Para recuperar a versão mais recente ARN da definição principal, use o `get-group-version` comando e especifique o ID da versão mais recente do grupo.

```
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"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Visão geral do modelo de objetos do grupo AWS IoT Greengrass no](#) Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateGroupVersion](#) na Referência de AWS CLI Comandos.

create-group

O código de exemplo a seguir mostra como usar create-group.

AWS CLI

Para criar um grupo Greengrass

O create-group exemplo a seguir cria um grupo chamado cli-created-group.

```
aws greengrass create-group \
  --name cli-created-group
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Visão geral do modelo de objetos do grupo AWS IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [CreateGroup](#) na Referência de AWS CLI Comandos.

create-logger-definition-version

O código de exemplo a seguir mostra como usar `create-logger-definition-version`.

AWS CLI

Para criar uma versão de definição do registrador

O `create-logger-definition-version` exemplo a seguir cria uma versão de definição de registrador e a associa a uma definição de registrador. A versão define quatro configurações de registro: 1) registros de componentes do sistema no sistema de arquivos do dispositivo principal, 2) registros de funções Lambda definidos pelo usuário no sistema de arquivos do dispositivo principal, 3) registros de componentes do sistema no Amazon Logs e 4) registros de funções Lambda definidos pelo usuário no Amazon CloudWatch Logs. CloudWatch Observação: para a integração do CloudWatch Logs, sua função de grupo deve conceder as permissões apropriadas.

```
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\\":\\"INF0\\",\\"Space\\":10240,\\"Type\\":\\"FileSystem\\"},{\\"Id\\":\\"3\\",
```

```
\\"Component\\":\\"GreengrassSystem\\",\\"Level\\":\\"WARN\\",\\"Type\\":\\"AWSCloudWatch\\"},
{"Id\\":\\"4\\",\\"Component\\":\\"Lambda\\",\\"Level\\":\\"INFO\\",\\"Type\\":\\"AWSCloudWatch
\\"}]"]"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Monitoramento com registros AWS do IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [CreateLoggerDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-logger-definition

O código de exemplo a seguir mostra como usar `create-logger-definition`.

AWS CLI

Para criar uma definição de registrador

O `create-logger-definition` exemplo a seguir cria uma definição de registrador que contém uma versão inicial da definição de registrador. A versão inicial define três configurações de registro: 1) registros de componentes do sistema no sistema de arquivos do dispositivo principal, 2) registros de funções Lambda definidos pelo usuário no sistema de arquivos do dispositivo principal e 3) registros de funções Lambda definidos pelo usuário no Amazon Logs. **CloudWatch Observação:** para a integração do CloudWatch Logs, sua função de grupo deve conceder as permissões apropriadas.

```
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\\"}]}"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Monitoramento com registros AWS do IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [CreateLoggerDefinition](#) na Referência de AWS CLI Comandos.

create-resource-definition-version

O código de exemplo a seguir mostra como usar `create-resource-definition-version`.

AWS CLI

Para criar uma versão de uma definição de recurso

O `create-resource-definition-version` exemplo a seguir cria uma nova versão de um `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:greengrass-TwilioAuthToken-ntS1p6\\"}}}]}"
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [CreateResourceDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-resource-definition

O código de exemplo a seguir mostra como usar `create-resource-definition`.

AWS CLI

Para criar uma definição de recurso

O `create-resource-definition` exemplo a seguir cria uma definição de recurso que contém uma lista de recursos a serem usados em um grupo do Greengrass. Neste exemplo, uma versão inicial da definição do recurso é incluída fornecendo uma lista de recursos. A lista inclui um recurso para um token de autorização do Twilio e outro ARN para um segredo armazenado no AWS Secrets Manager. Você deve criar o segredo antes de criar a definição do recurso.

```
aws greengrass create-resource-definition \
  --name MyGreengrassResources \
  --initial-version '{"Resources": [{"Id": "TwilioAuthToken
\',"Name": "MyTwilioAuthToken","\ResourceDataContainer":
{"SecretsManagerSecretResourceData": {"ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-TwilioAuthToken-ntSlp6\"]]}'"
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte [Como configurar o acesso a recursos locais usando a interface de linha de AWS comando](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateResourceDefinition](#) na Referência de AWS CLI Comandos.

create-software-update-job

O código de exemplo a seguir mostra como usar `create-software-update-job`.

AWS CLI

Para criar um trabalho de atualização de software para um núcleo

O `create-software-update-job` exemplo a seguir cria uma tarefa de atualização over-the-air (OTA) para atualizar o software AWS IoT Greengrass Core no núcleo cujo nome é `MyFirstGroup_Core`. Esse comando requer uma IAM função que permita o acesso aos pacotes de atualização de software no Amazon S3 e inclua `iot.amazonaws.com` como uma entidade confiável.

```

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

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [OTAAtualizações do software AWS IoT Greengrass Core no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [CreateSoftwareUpdateJob](#) na Referência de AWS CLI Comandos.

create-subscription-definition-version

O código de exemplo a seguir mostra como usar `create-subscription-definition-version`.

AWS CLI

Para criar uma nova versão de uma definição de assinatura

O `create-subscription-definition-version` exemplo a seguir cria uma nova versão de uma definição de assinatura que contém três assinaturas: uma notificação de acionamento, uma entrada de temperatura e um status de saída.

```

aws greengrass create-subscription-definition-version \
  --subscription-definition-id "9d611d57-5d5d-44bd-a3b4-feccbdd69112" \
  --subscriptions "[{"Id": "TriggerNotification", "Source":
  \"arn:aws:lambda:us-west-2:123456789012:function:TempMonitor:GG_TempMonitor
\", \"Subject\": \"twilio/txt\", \"Target\": \"arn:aws:greengrass:us-west-2:./
connectors/TwilioNotifications/versions/1\"}, {"Id": "TemperatureInput", \"Source
\": \"cloud\", \"Subject\": \"temperature/input\", \"Target\": \"arn:aws:lambda:us-
west-2:123456789012:function:TempMonitor:GG_TempMonitor\"}, {"Id": \"OutputStatus
\", \"Source\": \"arn:aws:greengrass:us-west-2:./connectors/TwilioNotifications/
versions/1\", \"Subject\": \"twilio/message/status\", \"Target\": \"cloud\"}]]"

```

Saída:

```

{
  "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"
}
```

- Para API obter detalhes, consulte [CreateSubscriptionDefinitionVersion](#) na Referência de AWS CLI Comandos.

create-subscription-definition

O código de exemplo a seguir mostra como usar `create-subscription-definition`.

AWS CLI

Para criar uma definição de assinatura

O `create-subscription-definition` exemplo a seguir cria uma definição de assinatura e especifica sua versão inicial. A versão inicial contém três assinaturas: uma para o MQTT tópico ao qual o conector se inscreve, uma para permitir que uma função receba leituras de temperatura da AWS IoT e uma para permitir que a IoT receba informações de AWS status do conector. O exemplo fornece o ARN alias da função Lambda que foi criado anteriormente usando o comando do 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\"}]}"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Introdução aos conectores \(CLI\)](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [CreateSubscriptionDefinition](#) na Referência de AWS CLI Comandos.

delete-connector-definition

O código de exemplo a seguir mostra como usar `delete-connector-definition`.

AWS CLI

Para excluir uma definição de conector

O `delete-connector-definition` exemplo a seguir exclui a definição especificada do conector Greengrass. Se você excluir uma definição de conector usada por um grupo, esse grupo não poderá ser implantado com êxito.

```
aws greengrass delete-connector-definition \
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteConnectorDefinition](#) na Referência de AWS CLI Comandos.

delete-core-definition

O código de exemplo a seguir mostra como usar `delete-core-definition`.

AWS CLI

Para excluir uma definição principal

O `delete-core-definition` exemplo a seguir exclui a definição básica especificada do Greengrass, incluindo todas as versões. Se você excluir um núcleo associado a um grupo do Greengrass, esse grupo não poderá ser implantado com êxito.

```
aws greengrass delete-core-definition \  
  --core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteCoreDefinition](#) na Referência de AWS CLI Comandos.

delete-device-definition

O código de exemplo a seguir mostra como usar delete-device-definition.

AWS CLI

Para excluir uma definição de dispositivo

O delete-device-definition exemplo a seguir exclui a definição do dispositivo especificado, incluindo todas as suas versões. Se você excluir uma versão de definição de dispositivo usada por uma versão de grupo, a versão de grupo não poderá ser implantada com êxito.

```
aws greengrass delete-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteDeviceDefinition](#) na Referência de AWS CLI Comandos.

delete-function-definition

O código de exemplo a seguir mostra como usar delete-function-definition.

AWS CLI

Para excluir uma definição de função

O delete-function-definition exemplo a seguir exclui a definição da função Greengrass especificada. Se você excluir uma definição de função usada por um grupo, esse grupo não poderá ser implantado com êxito.

```
aws greengrass delete-function-definition \  
  --function-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

```
--function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteFunctionDefinition](#) na Referência de AWS CLI Comandos.

delete-group

O código de exemplo a seguir mostra como usar delete-group.

AWS CLI

Para excluir um grupo

O delete-group exemplo a seguir exclui o grupo Greengrass especificado.

```
aws greengrass delete-group \  
  --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteGroup](#) na Referência de AWS CLI Comandos.

delete-logger-definition

O código de exemplo a seguir mostra como usar delete-logger-definition.

AWS CLI

Para excluir uma definição de registrador

O delete-logger-definition exemplo a seguir exclui a definição do registrador especificada, incluindo todas as versões da definição do registrador. Se você excluir uma versão de definição de registrador usada por uma versão de grupo, a versão de grupo não poderá ser implantada com êxito.

```
aws greengrass delete-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

Este comando não produz saída.

Para obter mais informações, consulte [Monitoramento com registros AWS do IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [DeleteLoggerDefinition](#) na Referência de AWS CLI Comandos.

delete-resource-definition

O código de exemplo a seguir mostra como usar `delete-resource-definition`.

AWS CLI

Para excluir uma definição de recurso

O `delete-resource-definition` exemplo a seguir exclui a definição de recurso especificada, incluindo todas as versões do recurso. Se você excluir uma definição de recurso usada por um grupo, esse grupo não poderá ser implantado com êxito.

```
aws greengrass delete-resource-definition \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteResourceDefinition](#) na Referência de AWS CLI Comandos.

delete-subscription-definition

O código de exemplo a seguir mostra como usar `delete-subscription-definition`.

AWS CLI

Para excluir uma definição de assinatura

O `delete-subscription-definition` exemplo a seguir exclui a definição de assinatura especificada do Greengrass. Se você excluir uma assinatura que está sendo usada por um grupo, esse grupo não poderá ser implantado com êxito.

```
aws greengrass delete-subscription-definition \  
  --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteSubscriptionDefinition](#) na Referência de AWS CLI Comandos.

disassociate-role-from-group

O código de exemplo a seguir mostra como usar `disassociate-role-from-group`.

AWS CLI

Para desassociar a função de um grupo do Greengrass

O `disassociate-role-from-group` exemplo a seguir dissocia a IAM função do grupo Greengrass especificado.

```
aws greengrass disassociate-role-from-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Saída:

```
{  
  "DisassociatedAt": "2019-09-10T20:05:49Z"  
}
```

Para obter mais informações, consulte [Configurar a função do grupo](#) no Guia do AWS desenvolvedor do IoT Greengrass.

- Para API obter detalhes, consulte [DisassociateRoleFromGroup](#) na Referência de AWS CLI Comandos.

disassociate-service-role-from-account

O código de exemplo a seguir mostra como usar `disassociate-service-role-from-account`.

AWS CLI

Para desassociar uma função de serviço da sua conta AWS

O `disassociate-service-role-from-account` exemplo a seguir remove a função de serviço associada à sua AWS conta. Se você não estiver usando a função de serviço

em nenhuma AWS região, use o `delete-role-policy` comando para separar a política `AWSGreengrassResourceAccessRolePolicy` gerenciada da função e, em seguida, use o `delete-role` comando para excluir a função.

```
aws greengrass disassociate-service-role-from-account
```

Saída:

```
{
  "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

Para obter mais informações, consulte [Greengrass Service Role](#) no IoT AWS Greengrass Developer Guide.

- Para API obter detalhes, consulte [DisassociateServiceRoleFromAccount](#) na Referência de AWS CLI Comandos.

get-associated-role

O código de exemplo a seguir mostra como usar `get-associated-role`.

AWS CLI

Para obter a função associada a um grupo do Greengrass

O `get-associated-role` exemplo a seguir obtém a IAM função associada ao grupo Greengrass especificado. A função de grupo é usada por funções e conectores locais do Lambda para acessar serviços. AWS

```
aws greengrass get-associated-role \
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Saída:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",
  "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

Para obter mais informações, consulte [Configurar a função do grupo](#) no Guia do AWS desenvolvedor do IoT Greengrass.

- Para API obter detalhes, consulte [GetAssociatedRole](#) na Referência de AWS CLI Comandos.

get-bulk-deployment-status

O código de exemplo a seguir mostra como usar `get-bulk-deployment-status`.

AWS CLI

Para verificar o status da sua implantação em massa

O `get-bulk-deployment-status` exemplo a seguir recupera as informações de status da operação de implantação em massa especificada. Neste exemplo, o arquivo que especificou os grupos a serem implantados tem um registro de entrada inválido.

```
aws greengrass get-bulk-deployment-status \
  --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Saída:

```
{
  "BulkDeploymentMetrics": {
    "InvalidInputRecords": 1,
    "RecordsProcessed": 1,
    "RetryAttempts": 0
  },
  "BulkDeploymentStatus": "Completed",
  "CreatedAt": "2019-06-25T16:11:33.265Z",
  "tags": {}
}
```

Para obter mais informações, consulte [Criar implantações em massa para grupos no Guia](#) do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [GetBulkDeploymentStatus](#) na Referência de AWS CLI Comandos.

get-connectivity-info

O código de exemplo a seguir mostra como usar `get-connectivity-info`.

AWS CLI

Para obter as informações de conectividade de um núcleo do Greengrass

O `get-connectivity-info` exemplo a seguir mostra os endpoints que os dispositivos podem usar para se conectar ao núcleo especificado do Greengrass. As informações de conectividade são uma lista de endereços IP ou nomes de domínio, com números de porta correspondentes e metadados opcionais definidos pelo cliente.

```
aws greengrass get-connectivity-info \  
  --thing-name "MyGroup_Core"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [GetConnectivityInfo](#) Referência de AWS CLI Comandos.

get-connector-definition-version

O código de exemplo a seguir mostra como usar `get-connector-definition-version`.

AWS CLI

Para recuperar informações sobre uma versão específica de uma definição de conector

O `get-connector-definition-version` exemplo a seguir recupera informações sobre a versão especificada da definição do conector especificado. Para recuperar todas as versões IDs da definição do conector, use o `list-connector-definition-versions` comando. Para recuperar o ID da última versão adicionada à definição do conector, use o `get-connector-definition` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-connector-definition-version \  
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \  
  --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
```

Saída:

```
{  
  "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"
```

```
}
```

Para obter mais informações, consulte [Integração com serviços e protocolos usando conectores do Greengrass no Guia do desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [GetConnectorDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-connector-definition

O código de exemplo a seguir mostra como usar `get-connector-definition`.

AWS CLI

Para recuperar informações sobre uma definição de conector

O `get-connector-definition` exemplo a seguir recupera informações sobre a definição do conector especificado. Para recuperar as definições IDs do seu conector, use o `list-connector-definitions` comando.

```
aws greengrass get-connector-definition \  
--connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Saída:

```
{  
  "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": {}  
}
```

Para obter mais informações, consulte [Integração com serviços e protocolos usando conectores do Greengrass no Guia do desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [GetConnectorDefinition](#) na Referência de AWS CLI Comandos.

get-core-definition-version

O código de exemplo a seguir mostra como usar `get-core-definition-version`.

AWS CLI

Para recuperar detalhes sobre uma versão específica da definição principal do Greengrass

O `get-core-definition-version` exemplo a seguir recupera informações sobre a versão especificada da definição principal especificada. Para recuperar todas as versões da definição principal, use o `list-core-definition-versions` comando. IDs Para recuperar o ID da última versão adicionada à definição principal, use o `get-core-definition` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-core-definition-version \
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46" \
  --core-definition-version-id "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"
```

Saída:

```
{
  "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeaac3-fd9d-4312-a8fd-ffa9404a20e0",
  "CreationTimestamp": "2019-06-18T16:21:21.351Z",
  "Definition": {
    "Cores": [
      {
        "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/928dea7b82331b47c3ff77b0e763fc5e64e2f7c884e6ef391baed9b6b8e21b45",
        "Id": "1a39aac7-0885-4417-91f6-23e4cea6c511",
        "SyncShadow": false,
        "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
GGGroup4Pi3_Core"
      }
    ]
  },
  "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
  "Version": "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"
```

```
}
```

- Para API obter detalhes, consulte [GetCoreDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-core-definition

O código de exemplo a seguir mostra como usar `get-core-definition`.

AWS CLI

Para recuperar detalhes de uma definição básica do Greengrass

O `get-core-definition` exemplo a seguir recupera informações sobre a definição de núcleo especificada. Para recuperar suas IDs definições principais, use o `list-core-definitions` comando.

```
aws greengrass get-core-definition \  
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"
```

Saída:

```
{  
  "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": {}  
}
```

- Para API obter detalhes, consulte [GetCoreDefinition](#) na Referência de AWS CLI Comandos.

get-deployment-status

O código de exemplo a seguir mostra como usar `get-deployment-status`.

AWS CLI

Para recuperar o status de uma implantação

O `get-deployment-status` exemplo a seguir recupera o status da implantação especificada do grupo Greengrass especificado. Para obter o ID de implantação, use o `list-deployments` comando e especifique o ID do grupo.

```
aws greengrass get-deployment-status \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

Saída:

```
{  
  "DeploymentStatus": "Success",  
  "DeploymentType": "NewDeployment",  
  "UpdatedAt": "2019-06-18T17:04:44.761Z"  
}
```

- Para API obter detalhes, consulte [GetDeploymentStatus](#) na Referência de AWS CLI Comandos.

get-device-definition-version

O código de exemplo a seguir mostra como usar `get-device-definition-version`.

AWS CLI

Para obter uma versão de definição de dispositivo

O `get-device-definition-version` exemplo a seguir recupera informações sobre a versão especificada da definição do dispositivo especificado. Para recuperar todas as versões IDs da definição do dispositivo, use o `list-device-definition-versions` comando. Para recuperar o ID da última versão adicionada à definição do dispositivo, use o `get-device-definition` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-device-definition-version \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \  
  --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"
```


Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetDeviceDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-device-definition

O código de exemplo a seguir mostra como usar `get-device-definition`.

AWS CLI

Para obter uma definição de dispositivo

O `get-device-definition` exemplo a seguir recupera informações sobre a definição do dispositivo especificado. Para recuperar as definições IDs do seu dispositivo, use o `list-device-definitions` comando.

```
aws greengrass get-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Saída:

```
{  
  "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"  
}
```

- Para API obter detalhes, consulte [GetDeviceDefinition](#) na Referência de AWS CLI Comandos.

get-function-definition-version

O código de exemplo a seguir mostra como usar `get-function-definition-version`.

AWS CLI

Para recuperar detalhes sobre uma versão específica de uma função Lambda

O seguinte `get-function-definition-version` recupera informações sobre a versão especificada da definição da função especificada. Para recuperar todas as versões IDs da definição da função, use o `list-function-definition-versions` comando. Para recuperar o ID da última versão adicionada à definição da função, use o `get-function-definition` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-function-definition-version \  
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \  
  --version-id "1"
```

```
--function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetFunctionDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-function-definition

O código de exemplo a seguir mostra como usar `get-function-definition`.

AWS CLI

Para recuperar uma definição de função

O `get-function-definition` exemplo a seguir exibe detalhes da definição da função especificada. Para recuperar suas definições IDs de função, use o `list-function-definitions` comando.

```
aws greengrass get-function-definition \
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

Saída:

```
{
  "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": {}
}
```

- Para API obter detalhes, consulte [GetFunctionDefinition](#) na Referência de AWS CLI Comandos.

get-group-certificate-authority

O código de exemplo a seguir mostra como usar `get-group-certificate-authority`.

AWS CLI

Para recuperar a CA associada a um grupo do Greengrass

O `get-group-certificate-authority` exemplo a seguir recupera a autoridade de certificação (CA) associada ao grupo Greengrass especificado. Para obter o ID da autoridade de

AWS CLI

Para recuperar a configuração da autoridade de certificação usada pelo grupo Greengrass

O `get-group-certificate-configuration` exemplo a seguir recupera a configuração da autoridade de certificação (CA) usada pelo grupo Greengrass especificado.

```
aws greengrass get-group-certificate-configuration \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Saída:

```
{  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "CertificateExpiryInMilliseconds": 604800000,  
  "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"  
}
```

- Para API obter detalhes, consulte [GetGroupCertificateConfiguration](#) na Referência de AWS CLI Comandos.

get-group-version

O código de exemplo a seguir mostra como usar `get-group-version`.

AWS CLI

Para recuperar informações sobre uma versão de um grupo do Greengrass

O `get-group-version` exemplo a seguir recupera informações sobre a versão especificada do grupo especificado. Para recuperar todas as versões do grupo, use o `list-group-versions` comando. IDs Para recuperar o ID da última versão adicionada ao grupo, use o `get-group` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-group-version \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

Saída:

```
{
```

```

    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731/versions/115136b3-cfd7-4462-
b77f-8741a4b00e5e",
    "CreationTimestamp": "2019-06-18T17:04:30.915Z",
    "Definition": {
      "CoreDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-
fd9d-4312-a8fd-ffa9404a20e0",
      "FunctionDefinitionVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/functions/063f5d1a-1dd1-40b4-9b51-56f8993d0f85/
versions/9748fda7-1589-4fcc-ac94-f5559e88678b",
      "SubscriptionDefinitionVersionArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/definition/subscriptions/70e49321-83d5-45d2-
bc09-81f4917ae152/versions/88ae8699-12ac-4663-ba3f-4d7f0519140b"
    },
    "Id": "1013db12-8b58-45ff-acc7-704248f66731",
    "Version": "115136b3-cfd7-4462-b77f-8741a4b00e5e"
  }
}

```

- Para API obter detalhes, consulte [GetGroupVersion](#) na Referência de AWS CLI Comandos.

get-group

O código de exemplo a seguir mostra como usar `get-group`.

AWS CLI

Para recuperar informações sobre um grupo do Greengrass

O `get-group` exemplo a seguir recupera informações sobre o grupo Greengrass especificado. Para recuperar o IDs dos seus grupos, use o `list-groups` comando.

```

aws greengrass get-group \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"

```

Saída:

```

{
  "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": {}
}

```

- Para API obter detalhes, consulte [GetGroup](#) na Referência de AWS CLI Comandos.

get-logger-definition-version

O código de exemplo a seguir mostra como usar `get-logger-definition-version`.

AWS CLI

Para recuperar informações sobre uma versão de uma definição de registrador

O `get-logger-definition-version` exemplo a seguir recupera informações sobre a versão especificada da definição do registrador especificado. Para recuperar todas as versões IDs da definição do registrador, use o `list-logger-definition-versions` comando. Para recuperar o ID da última versão adicionada à definição do registrador, use o `get-logger-definition` comando e verifique a `LatestVersion` propriedade.

```

aws greengrass get-logger-definition-version \
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \
  --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"

```

Saída:

```

{
  "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"
}

```



```
}
```

- Para API obter detalhes, consulte [GetLoggerDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-logger-definition

O código de exemplo a seguir mostra como usar `get-logger-definition`.

AWS CLI

Para recuperar informações sobre uma definição de registrador

O `get-logger-definition` exemplo a seguir recupera informações sobre a definição de registrador especificada. Para recuperar as definições IDs do seu registrador, use o `list-logger-definitions` comando.

```
aws greengrass get-logger-definition \  
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

Saída:

```
{  
  "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": {}  
}
```

- Para API obter detalhes, consulte [GetLoggerDefinition](#) na Referência de AWS CLI Comandos.

get-resource-definition-version

O código de exemplo a seguir mostra como usar `get-resource-definition-version`.

AWS CLI

Para recuperar informações sobre uma versão específica de uma definição de recurso

O `get-resource-definition-version` exemplo a seguir recupera informações sobre a versão especificada da definição do recurso especificado. Para recuperar todas as versões IDs da definição do recurso, use o `list-resource-definition-versions` comando. Para recuperar o ID da última versão adicionada à definição do recurso, use o `get-resource-definition` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-resource-definition-version \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
  --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetResourceDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-resource-definition

O código de exemplo a seguir mostra como usar `get-resource-definition`.

AWS CLI

Para recuperar informações sobre uma definição de recurso

O `get-resource-definition` exemplo a seguir recupera informações sobre a definição do recurso especificado. Para recuperar suas definições IDs de recursos, use o `list-resource-definitions` comando.

```
aws greengrass get-resource-definition \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Saída:

```
{  
  "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": {}  
}
```

- Para API obter detalhes, consulte [GetResourceDefinition](#) na Referência de AWS CLI Comandos.

get-service-role-for-account

O código de exemplo a seguir mostra como usar `get-service-role-for-account`.

AWS CLI

Para recuperar os detalhes da função de serviço que está anexada à sua conta

O `get-service-role-for-account` exemplo a seguir recupera informações sobre a função de serviço vinculada à sua AWS conta.

```
aws greengrass get-service-role-for-account
```

Saída:

```
{
  "AssociatedAt": "2018-10-18T15:59:20Z",
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Para obter mais informações, consulte [Greengrass Service Role](#) no IoT AWS Greengrass Developer Guide.

- Para API obter detalhes, consulte [GetServiceRoleForAccount](#) na Referência de AWS CLI Comandos.

`get-subscription-definition-version`

O código de exemplo a seguir mostra como usar `get-subscription-definition-version`.

AWS CLI

Para recuperar informações sobre uma versão específica de uma definição de assinatura

O `get-subscription-definition-version` exemplo a seguir recupera informações sobre a versão especificada da definição de assinatura especificada. Para recuperar todas as versões IDs da definição de assinatura, use o `list-subscription-definition-versions` comando. Para recuperar o ID da última versão adicionada à definição de assinatura, use o `get-subscription-definition` comando e verifique a `LatestVersion` propriedade.

```
aws greengrass get-subscription-definition-version \
  --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \
  --subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [GetSubscriptionDefinitionVersion](#) na Referência de AWS CLI Comandos.

get-subscription-definition

O código de exemplo a seguir mostra como usar `get-subscription-definition`.

AWS CLI

Para recuperar informações sobre uma definição de assinatura

O `get-subscription-definition` exemplo a seguir recupera informações sobre a definição de assinatura especificada. Para recuperar as definições IDs de sua assinatura, use o `list-subscription-definitions` comando.

```
aws greengrass get-subscription-definition \
  --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

Saída:

```
{
  "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": {}
}
```

- Para API obter detalhes, consulte [GetSubscriptionDefinition](#) na Referência de AWS CLI Comandos.

get-thing-runtime-configuration

O código de exemplo a seguir mostra como usar `get-thing-runtime-configuration`.

AWS CLI

Para recuperar a configuração de tempo de execução de um núcleo do Greengrass

O `get-thing-runtime-configuration` exemplo a seguir recupera a configuração de tempo de execução de um núcleo do Greengrass. Antes de recuperar a configuração de tempo de execução, você deve usar o `update-thing-runtime-configuration` comando para criar uma configuração de tempo de execução para o núcleo.

```
aws greengrass get-thing-runtime-configuration \
  --thing-name SampleGreengrassCore
```

Saída:

```
{
  "RuntimeConfiguration": {
    "TelemetryConfiguration": {
      "ConfigurationSyncStatus": "OutOfSync",
      "Telemetry": "On"
    }
  }
}
```

```
}
}
```

Para obter mais informações, consulte [Definindo as configurações de telemetria no Guia do Desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [GetThingRuntimeConfiguration](#) na Referência de AWS CLI Comandos.

list-bulk-deployment-detailed-reports

O código de exemplo a seguir mostra como usar `list-bulk-deployment-detailed-reports`.

AWS CLI

Para listar informações sobre implantações individuais em uma implantação em massa

O `list-bulk-deployment-detailed-reports` exemplo a seguir exibe informações sobre as implantações individuais em uma operação de implantação em massa, incluindo o status.

```
aws greengrass list-bulk-deployment-detailed-reports \
  --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

Saída:

```
{
  "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"
    },
    ...
]
}

```

Para obter mais informações, consulte [Criar implantações em massa para grupos no Guia](#) do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [ListBulkDeploymentDetailedReports](#) na Referência de AWS CLI Comandos.

list-bulk-deployments

O código de exemplo a seguir mostra como usar `list-bulk-deployments`.

AWS CLI

Para listar implantações em massa

O `list-bulk-deployments` exemplo a seguir lista todas as implantações em massa.

```
aws greengrass list-bulk-deployments
```

Saída:

```

{
  "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"
    }
  ]
}

```



```
}
```

Para obter mais informações, consulte [Criar implantações em massa para grupos no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [ListBulkDeployments](#) na Referência de AWS CLI Comandos.

list-connector-definition-versions

O código de exemplo a seguir mostra como usar `list-connector-definition-versions`.

AWS CLI

Para listar as versões que estão disponíveis para uma definição de conector

O `list-connector-definition-versions` exemplo a seguir lista as versões que estão disponíveis para a definição de conector especificada. Use o `list-connector-definitions` comando para obter o ID de definição do conector.

```
aws greengrass list-connector-definition-versions \  
--connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Integração com serviços e protocolos usando conectores do Greengrass no Guia do desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [ListConnectorDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-connector-definitions

O código de exemplo a seguir mostra como usar `list-connector-definitions`.

AWS CLI

Para listar os conectores do Greengrass que estão definidos

O `list-connector-definitions` exemplo a seguir lista todos os conectores do Greengrass definidos para sua conta. AWS

```
aws greengrass list-connector-definitions
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Integração com serviços e protocolos usando conectores do Greengrass no Guia do desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [ListConnectorDefinitions](#) na Referência de AWS CLI Comandos.

list-core-definition-versions

O código de exemplo a seguir mostra como usar `list-core-definition-versions`.

AWS CLI

Para listar as versões de uma definição básica do Greengrass

O `list-core-definitions` exemplo a seguir lista todas as versões da definição principal especificada do Greengrass. Você pode usar o `list-core-definitions` comando para obter o ID da versão.

```
aws greengrass list-core-definition-versions \  
  --core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListCoreDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-core-definitions

O código de exemplo a seguir mostra como usar `list-core-definitions`.

AWS CLI

Para listar as principais definições do Greengrass

O `list-core-definitions` exemplo a seguir lista todas as principais definições do Greengrass para sua AWS conta.

```
aws greengrass list-core-definitions
```

Saída:

```
{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [ListCoreDefinitions](#) na Referência de AWS CLI Comandos.

list-deployments

O código de exemplo a seguir mostra como usar `list-deployments`.

AWS CLI

Para listar as implantações de um grupo do Greengrass

O `list-deployments` exemplo a seguir lista as implantações do grupo Greengrass especificado. Você pode usar o `list-groups` comando para pesquisar seu ID de grupo.

```

aws greengrass list-deployments \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"

```

Saída:

```

{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListDeployments](#) na Referência de AWS CLI Comandos.

list-device-definition-versions

O código de exemplo a seguir mostra como usar `list-device-definition-versions`.

AWS CLI

Para listar as versões de uma definição de dispositivo

O `list-device-definition-versions` exemplo a seguir exibe as versões de definição de dispositivo associadas à definição de dispositivo especificada.

```
aws greengrass list-device-definition-versions \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListDeviceDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-device-definitions

O código de exemplo a seguir mostra como usar `list-device-definitions`.

AWS CLI

Para listar as definições do seu dispositivo

O `list-device-definitions` exemplo a seguir exibe detalhes sobre as definições do dispositivo em sua AWS conta na AWS região especificada.

```
aws greengrass list-device-definitions \  
  --region us-west-2
```

Saída:

```
{  
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [ListDeviceDefinitions](#) na Referência de AWS CLI Comandos.

list-function-definition-versions

O código de exemplo a seguir mostra como usar `list-function-definition-versions`.

AWS CLI

Para listar as versões de uma função Lambda

O `list-function-definition-versions` exemplo a seguir lista todas as versões da função Lambda especificada. Você pode usar o `list-function-definitions` comando para obter o ID.

```

aws greengrass list-function-definition-versions \
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"

```

Saída:

```

{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListFunctionDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-function-definitions

O código de exemplo a seguir mostra como usar `list-function-definitions`.

AWS CLI

Para listar as funções do Lambda

O `list-function-definitions` exemplo a seguir lista todas as funções do Lambda definidas para sua AWS conta.

aws greengrass list-function-definitions

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListFunctionDefinitions](#) na Referência de AWS CLI Comandos.

list-group-certificate-authorities

O código de exemplo a seguir mostra como usar `list-group-certificate-authorities`.

AWS CLI

Para listar o atual CAs de um grupo

O `list-group-certificate-authorities` exemplo a seguir lista as autoridades de certificação atuais (CAs) do grupo Greengrass especificado.

```

aws greengrass list-group-certificate-authorities \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"

```

Saída:

```

{
  "GroupCertificateAuthorities": [
    {
      "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
      "GroupCertificateAuthorityId":
"f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListGroupCertificateAuthorities](#) na Referência de AWS CLI Comandos.

list-group-versions

O código de exemplo a seguir mostra como usar `list-group-versions`.

AWS CLI

Para listar as versões de um grupo do Greengrass

O `list-group-versions` exemplo a seguir lista as versões do grupo Greengrass especificado.

```
aws greengrass list-group-versions \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Saída:

```
{  
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListGroupVersions](#) na Referência de AWS CLI Comandos.

list-groups

O código de exemplo a seguir mostra como usar `list-groups`.

AWS CLI

Para listar os grupos do Greengrass

O `list-groups` exemplo a seguir lista todos os grupos do Greengrass que estão definidos em sua AWS conta.

```
aws greengrass list-groups
```

Saída:

```
{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [ListGroups](#) na Referência de AWS CLI Comandos.

list-logger-definition-versions

O código de exemplo a seguir mostra como usar `list-logger-definition-versions`.

AWS CLI

Para obter uma lista das versões de uma definição de registrador

O `list-logger-definition-versions` exemplo a seguir obtém uma lista de todas as versões da definição de registrador especificada.

```
aws greengrass list-logger-definition-versions \  
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListLoggerDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-logger-definitions

O código de exemplo a seguir mostra como usar `list-logger-definitions`.

AWS CLI

Para obter uma lista das definições do registrador

O `list-logger-definitions` exemplo a seguir lista todas as definições de registrador da sua AWS conta.

```
aws greengrass list-logger-definitions
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListLoggerDefinitions](#) na Referência de AWS CLI Comandos.

list-resource-definition-versions

O código de exemplo a seguir mostra como usar `list-resource-definition-versions`.

AWS CLI

Para listar as versões de uma definição de recurso

O `list-resource-definition-versions` exemplo a seguir lista as versões do recurso Greengrass especificado.

```
aws greengrass list-resource-definition-versions \
```



```
--resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListResourceDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-resource-definitions

O código de exemplo a seguir mostra como usar `list-resource-definitions`.

AWS CLI

Para listar os recursos que estão definidos

O `list-resource-definitions` exemplo a seguir lista os recursos definidos para o AWS IoT Greengrass usar.

```
aws greengrass list-resource-definitions
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListResourceDefinitions](#) na Referência de AWS CLI Comandos.

list-subscription-definition-versions

O código de exemplo a seguir mostra como usar `list-subscription-definition-versions`.

AWS CLI

Para listar as versões de uma definição de assinatura

O `list-subscription-definition-versions` exemplo a seguir lista todas as versões da assinatura especificada. Você pode usar o `list-subscription-definitions` comando para pesquisar o ID da assinatura.

```
aws greengrass list-subscription-definition-versions \  
--subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListSubscriptionDefinitionVersions](#) na Referência de AWS CLI Comandos.

list-subscription-definitions

O código de exemplo a seguir mostra como usar `list-subscription-definitions`.

AWS CLI

Para obter uma lista de definições de assinatura

O `list-subscription-definitions` exemplo a seguir lista todas as assinaturas do AWS IoT Greengrass definidas em sua conta. AWS

```
aws greengrass list-subscription-definitions
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListSubscriptionDefinitions](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags anexadas a um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags e seus valores que estão anexados ao recurso especificado.

```
aws greengrass list-tags-for-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Saída:

```
{  
  "tags": {  
    "ResourceSubType": "USB",  
    "ResourceType": "Device"  
  }  
}
```

Para obter mais informações, consulte Como [marcar seus recursos do Greengrass](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

reset-deployments

O código de exemplo a seguir mostra como usar `reset-deployments`.

AWS CLI

Para limpar as informações de implantação de um grupo do Greengrass

O `reset-deployments` exemplo a seguir limpa as informações de implantação do grupo Greengrass especificado. Quando você adiciona o `--force` option, as informações de implantação são redefinidas sem esperar que o dispositivo principal responda.

```
aws greengrass reset-deployments \  
  --group-name "greengrass-group-name"
```

```
--group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
--force
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Redefinir implantações no Guia](#) do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [ResetDeployments](#) na Referência de AWS CLI Comandos.

start-bulk-deployment

O código de exemplo a seguir mostra como usar start-bulk-deployment.

AWS CLI

Para iniciar uma operação de implantação em massa

O start-bulk-deployment exemplo a seguir inicia uma operação de implantação em massa, usando um arquivo armazenado em um bucket do S3 para especificar os grupos a serem implantados.

```
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\"}"
```

Saída:

```
{  
  "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",  
  "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```



```
--resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
--tags "ResourceType=Device,ResourceSubType=USB"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos do Greengrass](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag e seu valor de um recurso

O `untag-resource` exemplo a seguir remove a tag cuja chave é `Category` do grupo Greengrass especificado. Se a chave `Category` não existir para o recurso especificado, nenhum erro será retornado.

```
aws greengrass untag-resource \
--resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731" \
--tag-keys "Category"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos do Greengrass](#) no Guia do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-connectivity-info

O código de exemplo a seguir mostra como usar `update-connectivity-info`.

AWS CLI

Para atualizar as informações de conectividade de um núcleo do Greengrass

O `update-connectivity-info` exemplo a seguir altera os endpoints que os dispositivos podem usar para se conectar ao núcleo especificado do Greengrass. As informações de conectividade são uma lista de endereços IP ou nomes de domínio, com números de porta correspondentes e metadados opcionais definidos pelo cliente. Talvez seja necessário atualizar as informações de conectividade quando a rede local mudar.

```
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\"}]"
```

Saída:

```
{  
  "Version": "312de337-59af-4cf9-a278-2a23bd39c300"  
}
```

- Para API obter detalhes, consulte [UpdateConnectivityInfo](#) na Referência de AWS CLI Comandos.

update-connector-definition

O código de exemplo a seguir mostra como usar `update-connector-definition`.

AWS CLI

Para atualizar o nome de uma definição de conector

O `update-connector-definition` exemplo a seguir atualiza o nome da definição do conector especificado. Se você quiser atualizar os detalhes do conector, use o `create-connector-definition-version` comando para criar uma nova versão.

```
aws greengrass update-connector-definition \  
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \  
  --name "GreengrassConnectors2019"
```

Para obter mais informações, consulte [Integração com serviços e protocolos usando conectores no Guia](#) do desenvolvedor do AWS IoT Greengrass.

- Para API obter detalhes, consulte [UpdateConnectorDefinition](#) na Referência de AWS CLI Comandos.

update-core-definition

O código de exemplo a seguir mostra como usar `update-core-definition`.

AWS CLI

Para atualizar uma definição principal

O `update-core-definition` exemplo a seguir altera o nome da definição principal especificada. Você pode atualizar somente a name propriedade de uma definição principal.

```
aws greengrass update-core-definition \  
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \  
  --name "MyCoreDevices"
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurar o AWS IoT Greengrass Core no Guia do Desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [UpdateCoreDefinition](#) na Referência de AWS CLI Comandos.

update-device-definition

O código de exemplo a seguir mostra como usar `update-device-definition`.

AWS CLI

Para atualizar a definição de um dispositivo

O `update-device-definition` exemplo a seguir altera o nome da definição do dispositivo especificado. Você só pode atualizar a name propriedade de uma definição de dispositivo.

```
aws greengrass update-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \  
  --name "TemperatureSensors"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateDeviceDefinition](#) na Referência de AWS CLI Comandos.

update-function-definition

O código de exemplo a seguir mostra como usar `update-function-definition`.

AWS CLI

Para atualizar o nome de uma definição de função

O `update-function-definition` exemplo a seguir atualiza o nome da definição da função especificada. Se você quiser atualizar os detalhes da função, use o `create-function-definition-version` comando para criar uma nova versão.

```
aws greengrass update-function-definition \  
  --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \  
  --name ObsoleteFunction
```

Este comando não produz saída.

Para obter mais informações, consulte [Executar funções Lambda locais](#) no Guia do desenvolvedor do IoT AWS Greengrass.

- Para API obter detalhes, consulte [UpdateFunctionDefinition](#) na Referência de AWS CLI Comandos.

update-group-certificate-configuration

O código de exemplo a seguir mostra como usar `update-group-certificate-configuration`.

AWS CLI

Para atualizar a expiração dos certificados de um grupo

O `update-group-certificate-configuration` exemplo a seguir define uma expiração de 10 dias para os certificados gerados para o grupo especificado.

```
aws greengrass update-group-certificate-configuration \  
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \  
  --expiration-days 10
```

```
--certificate-expiry-in-milliseconds 864000000
```

Saída:

```
{  
  "CertificateExpiryInMilliseconds": 864000000,  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"  
}
```

Para obter mais informações, consulte Segurança [AWS do IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [UpdateGroupCertificateConfiguration](#) na Referência de AWS CLI Comandos.

update-group

O código de exemplo a seguir mostra como usar update-group.

AWS CLI

Para atualizar o nome do grupo

O update-group exemplo a seguir atualiza o nome do grupo Greengrass especificado. Se você quiser atualizar os detalhes do grupo, use o create-group-version comando para criar uma nova versão.

```
aws greengrass update-group \  
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
  --name TestGroup4of6
```

Para obter mais informações, consulte [Configurar o AWS IoT Greengrass na IoT no Guia do AWS Desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [UpdateGroup](#) na Referência de AWS CLI Comandos.

update-logger-definition

O código de exemplo a seguir mostra como usar update-logger-definition.

AWS CLI

Para atualizar uma definição de registrador

O `update-logger-definition` exemplo a seguir altera o nome da definição do registrador especificado. Você só pode atualizar a `name` propriedade de uma definição de registrador.

```
aws greengrass update-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \  
  --name "LoggingConfigsForSensors"
```

Este comando não produz saída.

Para obter mais informações, consulte [Monitoramento com registros AWS do IoT Greengrass no Guia do desenvolvedor do AWS IoT Greengrass](#).

- Para API obter detalhes, consulte [UpdateLoggerDefinition](#) na Referência de AWS CLI Comandos.

update-resource-definition

O código de exemplo a seguir mostra como usar `update-resource-definition`.

AWS CLI

Para atualizar o nome de uma definição de recurso

O `update-resource-definition` exemplo a seguir atualiza o nome da definição do recurso especificado. Se você quiser alterar os detalhes do recurso, use o `create-resource-definition-version` comando para criar uma nova versão.

```
aws greengrass update-resource-definition \  
  --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \  
  --name GreengrassConnectorResources
```

Este comando não produz saída.

Para obter mais informações, consulte [Acesse recursos locais com funções e conectores Lambda no Guia do desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [UpdateResourceDefinition](#) na Referência de AWS CLI Comandos.

update-subscription-definition

O código de exemplo a seguir mostra como usar `update-subscription-definition`.

AWS CLI

Para atualizar o nome de uma definição de assinatura

O `update-subscription-definition` exemplo a seguir atualiza o nome da definição de assinatura especificada. Se você quiser alterar os detalhes da assinatura, use o `create-subscription-definition-version` comando para criar uma nova versão.

```
aws greengrass update-subscription-definition \  
  --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \  
  --name "ObsoleteSubscription"
```

Este comando não produz saída.

Para obter mais informações, consulte o título no guia.

- Para API obter detalhes, consulte [UpdateSubscriptionDefinition](#) na Referência de AWS CLI Comandos.

update-thing-runtime-configuration

O código de exemplo a seguir mostra como usar `update-thing-runtime-configuration`.

AWS CLI

Para ativar a telemetria na configuração de tempo de execução de um núcleo do Greengrass

O `update-thing-runtime-configuration` exemplo a seguir atualiza a configuração de tempo de execução de um núcleo do Greengrass para ativar a telemetria.

```
aws greengrass update-thing-runtime-configuration \  
  --thing-name SampleGreengrassCore \  
  --telemetry-configuration {"Telemetry\":"\n\"}
```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo as configurações de telemetria no Guia do Desenvolvedor](#) do AWS IoT Greengrass.

- Para API obter detalhes, consulte [UpdateThingRuntimeConfiguration](#) na Referência de AWS CLI Comandos.

AWS IoT Greengrass V2 exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Greengrass V2.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-service-role-to-account

O código de exemplo a seguir mostra como usar `associate-service-role-to-account`.

AWS CLI

Para associar a função de serviço do Greengrass à sua conta AWS

O `associate-service-role-to-account` exemplo a seguir associa uma função de serviço ao AWS IoT AWS Greengrass para sua conta.

```
aws greengrassv2 associate-service-role-to-account \
  --role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

Saída:

```
{
  "associatedAt": "2022-01-19T19:21:53Z"
```

```
}
```

Para obter mais informações, consulte a [função de serviço do Greengrass no Guia do desenvolvedor do IoT AWS Greengrass V2](#).

- Para API obter detalhes, consulte [AssociateServiceRoleToAccount](#) na Referência de AWS CLI Comandos.

batch-associate-client-device-with-core-device

O código de exemplo a seguir mostra como usar `batch-associate-client-device-with-core-device`.

AWS CLI

Para associar dispositivos cliente a um dispositivo principal

O `batch-associate-client-device-with-core-device` exemplo a seguir associa dois dispositivos cliente a um dispositivo principal.

```
aws greengrassv2 batch-associate-client-device-with-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Saída:

```
{
  "errorEntries": []
}
```

Para obter mais informações, consulte [Interaja com dispositivos de IoT locais](#) no Guia do desenvolvedor do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [BatchAssociateClientDeviceWithCoreDevice](#) na Referência de AWS CLI Comandos.

batch-disassociate-client-device-from-core-device

O código de exemplo a seguir mostra como usar `batch-disassociate-client-device-from-core-device`.

AWS CLI

Para dissociar dispositivos cliente de um dispositivo principal

O `batch-disassociate-client-device-from-core-device` exemplo a seguir dissocia dois dispositivos cliente de um dispositivo principal.

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \  
  --core-device-thing-name MyGreengrassCore \  
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Saída:

```
{  
  "errorEntries": []  
}
```

Para obter mais informações, consulte [Interaja com dispositivos de IoT locais](#) no Guia do desenvolvedor do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [BatchDisassociateClientDeviceFromCoreDevice](#) na Referência de AWS CLI Comandos.

cancel-deployment

O código de exemplo a seguir mostra como usar `cancel-deployment`.

AWS CLI

Para cancelar uma implantação

O `cancel-deployment` exemplo a seguir interrompe uma implantação contínua em um grupo de coisas.

```
aws greengrassv2 cancel-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "message": "SUCCESS"
```

```
}
```

Para obter mais informações, consulte [Cancelar implantações no Guia](#) do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [CancelDeployment](#) na Referência de AWS CLI Comandos.

create-component-version

O código de exemplo a seguir mostra como usar `create-component-version`.

AWS CLI

Exemplo 1: Para criar uma versão de componente a partir de uma receita

O `create-component-version` exemplo a seguir cria uma versão de um componente Hello World a partir de um arquivo de receita.

```
aws greengrassv2 create-component-version \  
--inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

Conteúdo de `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}'"  
      }  
    }  
  ]  
}
```

```
]
}
```

Saída:

```
{
  "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": {}
  }
}
```

Para obter mais informações, consulte [Criar componentes personalizados](#) e [Carregar componentes para implantação no Guia](#) do desenvolvedor do AWS IoT Greengrass V2.

Exemplo 2: Para criar uma versão de componente a partir de uma função AWS Lambda

O `create-component-version` exemplo a seguir cria uma versão de um componente Hello World a partir de uma função AWS Lambda.

```
aws greengrassv2 create-component-version \
  --cli-input-json file://lambda-function-component.json
```

Conteúdo de `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"
        }
      ]
    }
  }
}
```

```

    }
  ]
}
}
}

```

Saída:

```

{
  "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": {}
  }
}

```

Para obter mais informações, consulte [Executar funções do AWS Lambda no Guia](#) do desenvolvedor do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [CreateComponentVersion](#) na Referência de AWS CLI Comandos.

create-deployment

O código de exemplo a seguir mostra como usar create-deployment.

AWS CLI

Exemplo 1: Para criar uma implantação

O create-deployment exemplo a seguir implanta a interface de linha de comando do AWS IoT Greengrass em um dispositivo principal.

```

aws greengrassv2 create-deployment \
  --cli-input-json file://cli-deployment.json

```

Conteúdo de 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": {}
}
```

Saída:

```
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Criar implantações no Guia](#) do desenvolvedor do AWS IoT Greengrass V2.

Exemplo 2: Para criar uma implantação que atualize as configurações dos componentes

O `create-deployment` exemplo a seguir implanta o componente nuclear do AWS IoT Greengrass em um grupo de dispositivos principais. Essa implantação aplica as seguintes atualizações de configuração para o componente do núcleo:

Redefina as configurações de proxy dos dispositivos de destino para as configurações padrão sem proxy. Redefina as MQTT configurações dos dispositivos de destino para seus padrões. Define as JVM opções para o núcleo. Define o nível de registro do núcleo. JVM

```
aws greengrassv2 create-deployment \
```

```
--cli-input-json file://nucleus-deployment.json
```

Conteúdo de 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": {}
}
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Criar implantações](#) e [atualizar configurações de componentes no Guia do desenvolvedor](#) do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [CreateDeployment](#) na Referência de AWS CLI Comandos.

delete-component

O código de exemplo a seguir mostra como usar delete-component.

AWS CLI

Para excluir uma versão do componente

O delete-component exemplo a seguir exclui um componente Hello World.

```
aws greengrassv2 delete-component \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [DeleteComponent](#) na Referência de AWS CLI Comandos.

delete-core-device

O código de exemplo a seguir mostra como usar delete-core-device.

AWS CLI

Para excluir um dispositivo principal

O delete-core-device exemplo a seguir exclui um dispositivo principal do AWS IoT Greengrass.

```
aws greengrassv2 delete-core-device \  
  --core-device-thing-name MyGreengrassCore
```

Este comando não produz saída.

Para obter mais informações, consulte [Desinstalar o software AWS IoT Greengrass Core no Guia do desenvolvedor do AWS IoT Greengrass V2](#).

- Para API obter detalhes, consulte [DeleteCoreDevice](#) na Referência de AWS CLI Comandos.

describe-component

O código de exemplo a seguir mostra como usar describe-component.

AWS CLI

Para descrever a versão de um componente

O describe-component exemplo a seguir descreve um componente Hello World.

```
aws greengrassv2 describe-component \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

Saída:

```
{  
  "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"  
      }  
    }  
  ]  
}
```


Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [DescribeComponent](#) na Referência de AWS CLI Comandos.

disassociate-service-role-from-account

O código de exemplo a seguir mostra como usar `disassociate-service-role-from-account`.

AWS CLI

Para desassociar a função de serviço do Greengrass da sua conta AWS

O `disassociate-service-role-from-account` exemplo a seguir dissocia a função de serviço do Greengrass do AWS IoT Greengrass para sua conta. AWS

```
aws greengrassv2 disassociate-service-role-from-account
```

Saída:

```
{
  "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

Para obter mais informações, consulte a [função de serviço do Greengrass no Guia](#) do desenvolvedor do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [DisassociateServiceRoleFromAccount](#) na Referência de AWS CLI Comandos.

get-component-version-artifact

O código de exemplo a seguir mostra como usar `get-component-version-artifact`.

AWS CLI

Para obter um URL para baixar um artefato de componente

O `get-component-version-artifact` exemplo a seguir obtém um URL para baixar o JAR arquivo do componente do console de depuração local.

```
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"
```

Saída:

```
{
  "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..."
}
```

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [GetComponentVersionArtifact](#) na Referência de AWS CLI Comandos.

get - component

O código de exemplo a seguir mostra como usar `get - component`.

AWS CLI

Exemplo 1: Para baixar a receita de um componente em YAML formato (Linux, macOS ou Unix)

O `get - component` exemplo a seguir baixa a receita de um componente Hello World em um arquivo em YAML formato. Esse comando faz o seguinte:

Usa os `--query` parâmetros `--output` e para controlar a saída do comando. Esses parâmetros extraem o blob da receita da saída do comando. Para obter mais informações sobre como controlar a saída, consulte [Controlando a saída do comando](#) no Guia do usuário da interface de linha de AWS comando. Usa o `base64` utilitário. Esse utilitário decodifica o blob extraído para o texto original. O blob retornado por um `get - component` comando bem-sucedido é um texto codificado em base64. Você deve decodificar esse blob para obter o texto original. Salva o texto decodificado em um arquivo. A seção final do comando (`> com.example.HelloWorld-1.0.0.json`) salva o texto decodificado em um arquivo.

```
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
```

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

Exemplo 2: Para baixar a receita de um componente em YAML formato (WindowsCMD)

O `get-component` exemplo a seguir baixa a receita de um componente Hello World em um arquivo em YAML formato. Esse comando usa o `certutil` utilitário.

```
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
```

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

Exemplo 3: Para baixar a receita de um componente em YAML formato (Windows PowerShell)

O `get-component` exemplo a seguir baixa a receita de um componente Hello World em um arquivo em YAML formato. Esse comando usa o `certutil` utilitário.

```
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
```

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [GetComponent](#) na Referência de AWS CLI Comandos.

get-connectivity-info

O código de exemplo a seguir mostra como usar `get-connectivity-info`.

AWS CLI

Para obter as informações de conectividade de um dispositivo principal do Greengrass

O `get-connectivity-info` exemplo a seguir obtém as informações de conectividade de um dispositivo principal do Greengrass. Os dispositivos cliente usam essas informações para se conectar ao MQTT agente que é executado nesse dispositivo principal.

```
aws greengrassv2 get-connectivity-info \  
  --thing-name MyGreengrassCore
```

Saída:

```
{  
  "connectivityInfo": [  
    {  
      "id": "localIP_192.0.2.0",  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciar endpoints de dispositivos principais no Guia](#) do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [GetConnectivityInfo](#) na Referência de AWS CLI Comandos.

get-core-device

O código de exemplo a seguir mostra como usar `get-core-device`.

AWS CLI

Para obter um dispositivo principal

O `get-core-device` exemplo a seguir obtém informações sobre um dispositivo principal AWS do IoT Greengrass.

```
aws greengrassv2 get-core-device \  
  --core-device-thing-name MyGreengrassCore
```

Saída:

```
{  
  "coreDeviceThingName": "MyGreengrassCore",  
  "coreVersion": "2.0.3",  
  "platform": "linux",  
  "architecture": "amd64",  
  "status": "HEALTHY",  
  "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",  
  "tags": {}  
}
```

Para obter mais informações, consulte [Verificar o status do dispositivo principal](#) no Guia do AWS desenvolvedor do IoT Greengrass V2.

- Para API obter detalhes, consulte [GetCoreDevice](#) na Referência de AWS CLI Comandos.

get-deployment

O código de exemplo a seguir mostra como usar `get-deployment`.

AWS CLI

Para obter uma implantação

O `get-deployment` exemplo a seguir obtém informações sobre a implantação do componente do núcleo AWS IoT Greengrass em um grupo de dispositivos principais.

```
aws greengrassv2 get-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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": {}
}
```

Para obter mais informações, consulte [Implantar componentes em dispositivos](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [GetDeployment](#) na Referência de AWS CLI Comandos.

get-service-role-for-account

O código de exemplo a seguir mostra como usar `get-service-role-for-account`.

AWS CLI

Para obter a função de serviço do Greengrass em sua conta AWS

O `get-service-role-for-account` exemplo a seguir obtém a função de serviço associada ao AWS IoT Greengrass para sua conta. AWS

```
aws greengrassv2 get-service-role-for-account
```

Saída:

```
{
  "associatedAt": "2022-01-19T19:21:53Z",
  "roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Para obter mais informações, consulte a [função de serviço do Greengrass no Guia](#) do desenvolvedor do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [GetServiceRoleForAccount](#) na Referência de AWS CLI Comandos.

list-client-devices-associated-with-core-device

O código de exemplo a seguir mostra como usar `list-client-devices-associated-with-core-device`.

AWS CLI

Para listar os dispositivos cliente associados a um dispositivo principal

O `list-client-devices-associated-with-core-device` exemplo a seguir lista todos os dispositivos cliente associados a um dispositivo principal.

```
aws greengrassv2 list-client-devices-associated-with-core-device \
```

```
--core-device-thing-name MyTestGreengrassCore
```

Saída:

```
{
  "associatedClientDevices": [
    {
      "thingName": "MyClientDevice2",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    },
    {
      "thingName": "MyClientDevice1",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    }
  ]
}
```

Para obter mais informações, consulte [Interaja com dispositivos de IoT locais](#) no Guia do desenvolvedor do IoT AWS Greengrass V2.

- Para API obter detalhes, consulte [ListClientDevicesAssociatedWithCoreDevice](#) na Referência de AWS CLI Comandos.

list-component-versions

O código de exemplo a seguir mostra como usar `list-component-versions`.

AWS CLI

Para listar as versões de um componente

O `list-component-versions` exemplo a seguir lista todas as versões de um componente Hello World.

```
aws greengrassv2 list-component-versions \
  --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example>HelloWorld
```

Saída:

```
{
```



```
"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"  
  }  
]
```

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [ListComponentVersions](#) na Referência de AWS CLI Comandos.

list-components

O código de exemplo a seguir mostra como usar `list-components`.

AWS CLI

Para listar componentes

O `list-components` exemplo a seguir lista cada componente e sua versão mais recente definida em sua AWS conta na região atual.

```
aws greengrassv2 list-components
```

Saída:

```
{  
  "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"
          }
        }
      ]
    }
  ]
}

```

Para obter mais informações, consulte [Gerenciar componentes](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [ListComponents](#) na Referência de AWS CLI Comandos.

list-core-devices

O código de exemplo a seguir mostra como usar `list-core-devices`.

AWS CLI

Para listar os dispositivos principais

O `list-core-devices` exemplo a seguir lista os principais dispositivos do AWS IoT Greengrass em sua AWS conta na região atual.

```
aws greengrassv2 list-core-devices
```

Saída:

```
{
```

```
"coreDevices": [  
  {  
    "coreDeviceThingName": "MyGreengrassCore",  
    "status": "HEALTHY",  
    "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"  
  }  
]  
}
```

Para obter mais informações, consulte [Verificar o status do dispositivo principal](#) no Guia do AWS desenvolvedor do IoT Greengrass V2.

- Para API obter detalhes, consulte [ListCoreDevices](#) na Referência de AWS CLI Comandos.

list-deployments

O código de exemplo a seguir mostra como usar `list-deployments`.

AWS CLI

Para listar implantações

O `list-deployments` exemplo a seguir lista a revisão mais recente de cada implantação definida em sua AWS conta na região atual.

```
aws greengrassv2 list-deployments
```

Saída:

```
{  
  "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
    }
  ]
}

```

Para obter mais informações, consulte [Implantar componentes em dispositivos](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [ListDeployments](#) na Referência de AWS CLI Comandos.

list-effective-deployments

O código de exemplo a seguir mostra como usar `list-effective-deployments`.

AWS CLI

Para listar trabalhos de implantação

O `list-effective-deployments` exemplo a seguir lista as implantações que se aplicam a um dispositivo principal do AWS IoT Greengrass.

```

aws greengrassv2 list-effective-deployments \
  --core-device-thing-name MyGreengrassCore

```

Saída:

```

{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Verificar o status do dispositivo principal](#) no Guia do AWS desenvolvedor do IoT Greengrass V2.

- Para API obter detalhes, consulte [ListEffectiveDeployments](#) na Referência de AWS CLI Comandos.

list-installed-components

O código de exemplo a seguir mostra como usar `list-installed-components`.

AWS CLI

Para listar os componentes instalados em um dispositivo principal

O `list-installed-components` exemplo a seguir lista os componentes que estão instalados em um dispositivo principal AWS do IoT Greengrass.

```

aws greengrassv2 list-installed-components \
  --core-device-thing-name MyGreengrassCore

```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Verificar o status do dispositivo principal](#) no Guia do AWS desenvolvedor do IoT Greengrass V2.

- Para API obter detalhes, consulte [ListInstalledComponents](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um dispositivo AWS principal do IoT Greengrass.

```
aws greengrassv2 list-tags-for-resource \
  --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore
```

Saída:

```
{
```

```
"tags": {  
  "Owner": "richard-roe"  
}  
}
```

Para obter mais informações, consulte [Marcar seus recursos](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a um recurso

O `tag-resource` exemplo a seguir adiciona uma tag de proprietário a um dispositivo AWS principal do IoT Greengrass. Você pode usar essa tag para controlar o acesso ao dispositivo principal com base em quem o possui.

```
aws greengrassv2 tag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tags Owner=richard-roe
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar seus recursos](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove uma tag de proprietário de um dispositivo AWS principal do IoT Greengrass.

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tag-keys Owner
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar seus recursos](#) no Guia do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-connectivity-info

O código de exemplo a seguir mostra como usar `update-connectivity-info`.

AWS CLI

Para atualizar as informações de conectividade de um dispositivo principal do Greengrass

O `update-connectivity-info` exemplo a seguir obtém as informações de conectividade de um dispositivo principal do Greengrass. Os dispositivos cliente usam essas informações para se conectar ao MQTT agente que é executado nesse dispositivo principal.

```
aws greengrassv2 update-connectivity-info \  
  --thing-name MyGreengrassCore \  
  --cli-input-json file://core-device-connectivity-info.json
```

Conteúdo de `core-device-connectivity-info.json`:

```
{  
  "connectivityInfo": [  
    {  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883,  
      "id": "localIP_192.0.2.0"  
    }  
  ]  
}
```



```
]
}
```

Saída:

```
{
  "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Gerenciar endpoints de dispositivos principais no Guia](#) do desenvolvedor do AWS IoT Greengrass V2.

- Para API obter detalhes, consulte [UpdateConnectivityInfo](#) na Referência de AWS CLI Comandos.

AWS IoT Jobs SDK release exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Jobs SDK release.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-job-execution

O código de exemplo a seguir mostra como usar `describe-job-execution`.

AWS CLI

Para obter os detalhes da execução de um trabalho

O `describe-job-execution` exemplo a seguir recupera os detalhes da execução mais recente do trabalho e do item especificados.

```
aws iot-jobs-data describe-job-execution \  
  --job-id SampleJob \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Saída:

```
{  
  "execution": {  
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567701875.743,  
    "queuedAt": 1567701902.444,  
    "status": "QUEUED",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

Para obter mais informações, consulte [Dispositivos e trabalhos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DescribeJobExecution](#) na Referência de AWS CLI Comandos.

get-pending-job-executions

O código de exemplo a seguir mostra como usar `get-pending-job-executions`.

AWS CLI

Para obter uma lista de todos os trabalhos que não estão no status de terminal de uma coisa

O `get-pending-job-executions` exemplo a seguir exibe uma lista de todos os trabalhos que não estão em um estado terminal para o item especificado.

```
aws iot-jobs-data get-pending-job-executions \  
  --thing-name MotionSensor1
```

```
--endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Saída:

```
{
  "InProgressJobs": [
  ],
  "queuedJobs": [
    {
      "executionNumber": 2939653338,
      "jobId": "SampleJob",
      "lastUpdatedAt": 1567701875.743,
      "queuedAt": 1567701902.444,
      "versionNumber": 3
    }
  ]
}
```

Para obter mais informações, consulte [Dispositivos e trabalhos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetPendingJobExecutions](#) na Referência de AWS CLI Comandos.

start-next-pending-job-execution

O código de exemplo a seguir mostra como usar `start-next-pending-job-execution`.

AWS CLI

Para obter e iniciar a próxima execução de trabalho pendente de uma coisa

O `start-next-pending-job-execution` exemplo a seguir recupera e inicia a próxima execução do trabalho cujo status é `IN_PROGRESS` ou `QUEUED` para o item especificado.

```
aws iot-jobs-data start-next-pending-job-execution \
  --thing-name MotionSensor1
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Saída:

```
{
```

```
"execution": {
  "approximateSecondsBeforeTimedOut": 88,
  "executionNumber": 2939653338,
  "jobId": "SampleJob",
  "lastUpdatedAt": 1567714853.743,
  "queuedAt": 1567701902.444,
  "startedAt": 1567714871.690,
  "status": "IN_PROGRESS",
  "thingName": "MotionSensor1 ",
  "versionNumber": 3
}
```

Para obter mais informações, consulte [Dispositivos e trabalhos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [StartNextPendingJobExecution](#) na Referência de AWS CLI Comandos.

update-job-execution

O código de exemplo a seguir mostra como usar update-job-execution.

AWS CLI

Para atualizar o status da execução de um trabalho

O update-job-execution exemplo a seguir atualiza o status do trabalho e do item especificados.

```
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
```

Saída:

```
{
  "executionState": {
    "status": "REMOVED",
    "versionNumber": 3
  }
}
```

```
    },  
  }
```

Para obter mais informações, consulte [Dispositivos e trabalhos](#) no Guia do desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateJobExecution](#) na Referência de AWS CLI Comandos.

AWS IoT SiteWise exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT SiteWise.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-assets

O código de exemplo a seguir mostra como usar `associate-assets`.

AWS CLI

Para associar um ativo secundário a um ativo principal

O `associate-assets` exemplo a seguir associa um ativo de turbina eólica a um ativo de parque eólico, onde o modelo de ativo de turbina eólica existe como uma hierarquia no modelo de ativo de parque eólico.

```
aws iotsitewise associate-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \  
  --principal-asset-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --secondary-asset-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \  
  --role-arn arn:aws:iam::123456789012:role/IAMRoleName \  
  --region us-east-1 \  
  --profile my-profile
```

```
--child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Associando ativos no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [AssociateAssets](#) na Referência de AWS CLI Comandos.

batch-associate-project-assets

O código de exemplo a seguir mostra como usar `batch-associate-project-assets`.

AWS CLI

Para associar um ativo a um projeto

O `batch-associate-project-assets` exemplo a seguir associa um ativo de parque eólico a um projeto.

```
aws iotsitewise batch-associate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar ativos a projetos](#) no Guia de aplicativos do AWS IoT SiteWise Monitor.

- Para API obter detalhes, consulte [BatchAssociateProjectAssets](#) na Referência de AWS CLI Comandos.

batch-disassociate-project-assets

O código de exemplo a seguir mostra como usar `batch-disassociate-project-assets`.

AWS CLI

Para desassociar um ativo de um projeto

O `batch-disassociate-project-assets` exemplo a seguir dissocia um ativo de parque eólico de um projeto.

```
aws iotsitewise batch-disassociate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar ativos a projetos](#) no Guia de aplicativos do AWS IoT SiteWise Monitor.

- Para API obter detalhes, consulte [BatchDisassociateProjectAssets](#) na Referência de AWS CLI Comandos.

batch-put-asset-property-value

O código de exemplo a seguir mostra como usar batch-put-asset-property-value.

AWS CLI

Para enviar dados às propriedades do ativo

O batch-put-asset-property-value exemplo a seguir envia dados de potência e temperatura para as propriedades do ativo identificadas pelos aliases da propriedade.

```
aws iotsitewise batch-put-asset-property-value \  
  --cli-input-json file://batch-put-asset-property-value.json
```

Conteúdo de 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
      }
    }
  ]
}
]
}

```

Saída:

```

{
  "errorEntries": []
}

```

Para obter mais informações, consulte [Ingestão de dados usando a AWS SiteWise API IoT no Guia do usuário da AWS IoT SiteWise](#) .

- Para API obter detalhes, consulte [BatchPutAssetPropertyValue](#) na Referência de AWS CLI Comandos.

create-access-policy

O código de exemplo a seguir mostra como usar `create-access-policy`.

AWS CLI

Exemplo 1: Para conceder a um usuário acesso administrativo a um portal

O `create-access-policy` exemplo a seguir cria uma política de acesso que concede ao usuário acesso administrativo a um portal da web para uma empresa de parques eólicos.


```
aws iotsitewise create-access-policy \  
  --cli-input-json file://create-portal-administrator-access-policy.json
```

Conteúdo de 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"  
    }  
  }  
}
```

Saída:

```
{  
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/  
a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE"  
}
```

Para obter mais informações, consulte [Adicionando ou removendo administradores do portal](#) no Guia do Usuário de AWS SiteWise IoT.

Exemplo 2: Para conceder a um usuário acesso somente de leitura a um projeto

O create-access-policy exemplo a seguir cria uma política de acesso que concede ao usuário acesso somente de leitura a um projeto de parque eólico.

```
aws iotsitewise create-access-policy \  
  --cli-input-json file://create-project-viewer-access-policy.json
```

Conteúdo de create-project-viewer-access-policy.json:

```
{
```

```

    "accessPolicyIdentity": {
      "user": {
        "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
      }
    },
    "accessPolicyPermission": "VIEWER",
    "accessPolicyResource": {
      "project": {
        "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"
      }
    }
  }
}

```

Saída:

```

{
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"
}

```

Para obter mais informações, consulte [Atribuição de visualizadores de projetos](#) no Guia de aplicativos do AWS SiteWise IoT Monitor.

- Para API obter detalhes, consulte [CreateAccessPolicy](#) na Referência de AWS CLI Comandos.

create-asset-model

O código de exemplo a seguir mostra como usar `create-asset-model`.

AWS CLI

Para criar um modelo de ativo

O `create-asset-model` exemplo a seguir cria um modelo de ativo que define uma turbina eólica com as seguintes propriedades:

Número de série - O número de série de uma turbina
Generated energia eólica - O fluxo de dados de energia gerado por um vento
turbineTemperature C - O fluxo de dados de temperatura de uma turbina eólica em Celsius
Temperature F - Os pontos de dados de temperatura mapeados de Celsius a Fahrenheit

```
aws iotsitewise create-asset-model \  
  --cli-input-json file://create-wind-turbine-model.json
```

Conteúdo de 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"
          }
        }
      }
    }
  }
]
}

```

Saída:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Para obter mais informações, consulte [Definição de modelos de ativos](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [CreateAssetModel](#) na Referência de AWS CLI Comandos.

create-asset

O código de exemplo a seguir mostra como usar create-asset.

AWS CLI

Para criar um ativo

O create-asset exemplo a seguir cria um ativo de turbina eólica a partir de um modelo de ativo de turbina eólica.

```
aws iotsitewise create-asset \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --asset-name "Wind Turbine 1"
```

Saída:

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",  
  "assetStatus": {  
    "state": "CREATING"  
  }  
}
```

Para obter mais informações, consulte [Criação de ativos](#) no Guia do SiteWise Usuário de AWS IoT.

- Para API obter detalhes, consulte [CreateAsset](#) na Referência de AWS CLI Comandos.

create-dashboard

O código de exemplo a seguir mostra como usar create-dashboard.

AWS CLI

Para criar um painel

O `create-dashboard` exemplo a seguir cria um painel com um gráfico de linhas que exibe a potência total gerada para um parque eólico.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --dashboard-name "Wind Farm" \  
  --dashboard-definition file://create-wind-farm-dashboard.json
```

Conteúdo de `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"  
        }  
      ]  
    }  
  ]  
}
```

Saída:

```
{  
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",  
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/  
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE"  
}
```

Para obter mais informações, consulte [Criação de painéis \(CLI\) no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [CreateDashboard](#) na Referência de AWS CLI Comandos.

create-gateway

O código de exemplo a seguir mostra como usar create-gateway.

AWS CLI

Para criar um gateway

O create-gateway exemplo a seguir cria um gateway que é executado no 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}
```

Saída:

```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"  
}
```

Para obter mais informações, consulte [Configurando um gateway no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [CreateGateway](#) na Referência de AWS CLI Comandos.

create-portal

O código de exemplo a seguir mostra como usar create-portal.

AWS CLI

Para criar um portal

O `create-portal` exemplo a seguir cria um portal da web para uma empresa de parques eólicos. Você pode criar portais somente na mesma região em que habilitou o 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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Introdução ao SiteWise Monitor de AWS IoT](#) no Guia do Usuário de AWS IoT e Ativação [no AWS SSO](#) Guia SiteWise do Usuário de AWS IoT. SiteWise

- Para API obter detalhes, consulte [CreatePortal](#) na Referência de AWS CLI Comandos.

create-project

O código de exemplo a seguir mostra como usar `create-project`.

AWS CLI

Para criar um projeto

O `create-project` exemplo a seguir cria um projeto de parque eólico.

```
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."
```

Saída:

```
{  
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/  
a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"  
}
```

Para obter mais informações, consulte [Criação de projetos](#) no Guia de aplicativos do AWS IoT SiteWise Monitor.

- Para API obter detalhes, consulte [CreateProject](#) na Referência de AWS CLI Comandos.

delete-access-policy

O código de exemplo a seguir mostra como usar `delete-access-policy`.

AWS CLI

Para revogar o acesso de um usuário a um projeto ou portal

O `delete-access-policy` exemplo a seguir exclui uma política de acesso que concede ao usuário acesso administrativo a um portal.

```
aws iotsitewise delete-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionando ou removendo administradores do portal](#) no Guia do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DeleteAccessPolicy](#) na Referência de AWS CLI Comandos.

delete-asset-model

O código de exemplo a seguir mostra como usar `delete-asset-model`.

AWS CLI

Para excluir um modelo de ativo

O `delete-asset-model` exemplo a seguir exclui um modelo de ativo de turbina eólica.

```
aws iotsitewise delete-asset-model \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Saída:

```
{  
  "assetModelStatus": {  
    "state": "DELETING"  
  }  
}
```

Para obter mais informações, consulte [Excluindo modelos de ativos](#) no Guia do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DeleteAssetModel](#) na Referência de AWS CLI Comandos.

delete-asset

O código de exemplo a seguir mostra como usar `delete-asset`.

AWS CLI

Para excluir um ativo

O `delete-asset` exemplo a seguir exclui um ativo de turbina eólica.

```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Saída:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

Para obter mais informações, consulte [Excluindo ativos no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DeleteAsset](#) na Referência de AWS CLI Comandos.

delete-dashboard

O código de exemplo a seguir mostra como usar delete-dashboard.

AWS CLI

Para excluir um painel

O delete-dashboard exemplo a seguir exclui o painel de controle de uma turbina eólica.

```
aws iotsitewise delete-dashboard \  
  --dashboard-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo painéis](#) no Guia de aplicativos do AWS IoT Monitor SiteWise .

- Para API obter detalhes, consulte [DeleteDashboard](#) na Referência de AWS CLI Comandos.

delete-gateway

O código de exemplo a seguir mostra como usar delete-gateway.

AWS CLI

Para excluir um gateway

O delete-gateway exemplo a seguir exclui um gateway.

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte Como [ingerir dados usando um gateway no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DeleteGateway](#) na Referência de AWS CLI Comandos.

delete-portal

O código de exemplo a seguir mostra como usar `delete-portal`.

AWS CLI

Para excluir um portal

O `delete-portal` exemplo a seguir exclui um portal da web de uma empresa de parque eólico.

```
aws iotsitewise delete-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE
```

Saída:

```
{  
  "portalStatus": {  
    "state": "DELETING"  
  }  
}
```

Para obter mais informações, consulte [Excluindo um portal no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DeletePortal](#) na Referência de AWS CLI Comandos.

delete-project

O código de exemplo a seguir mostra como usar `delete-project`.

AWS CLI

Para excluir um projeto

O `delete-project` exemplo a seguir exclui um projeto de parque eólico.

```
aws iotsitewise delete-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo projetos no Guia](#) de aplicativos do AWS SiteWise IoT Monitor.

- Para API obter detalhes, consulte [DeleteProject](#) na Referência de AWS CLI Comandos.

describe-access-policy

O código de exemplo a seguir mostra como usar `describe-access-policy`.

AWS CLI

Para descrever uma política de acesso

O `describe-access-policy` exemplo a seguir descreve uma política de acesso que concede ao usuário acesso administrativo a um portal da web de uma empresa de parques eólicos.

```
aws iotsitewise describe-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Adicionando ou removendo administradores do portal](#) no Guia do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DescribeAccessPolicy](#) na Referência de AWS CLI Comandos.

describe-asset-model

O código de exemplo a seguir mostra como usar `describe-asset-model`.

AWS CLI

Para descrever um modelo de ativo

O `describe-asset-model` exemplo a seguir descreve um modelo de ativo de parque eólico.

```
aws iotsitewise describe-asset-model \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Saída:

```
{  
  "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"
}
}

```

Para obter mais informações, consulte [Descrição de um modelo de ativo específico](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [DescribeAssetModel](#) na Referência de AWS CLI Comandos.

describe-asset-property

O código de exemplo a seguir mostra como usar `describe-asset-property`.

AWS CLI

Para descrever uma propriedade do ativo

O `describe-asset-property` exemplo a seguir descreve a propriedade total de energia gerada de um ativo de parque eólico.

```
aws iotsitewise describe-asset-property \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE
```

Saída:

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",  
  "assetName": "Wind Farm 1",  
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",  
  "assetProperty": {  
    "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",  
    "name": "Total Generated Power",  
    "notification": {  
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",  
      "state": "DISABLED"  
    },  
    "dataType": "DOUBLE",  
    "unit": "kW",  
    "type": {  
      "metric": {  
        "expression": "sum(power)",  
        "variables": [  
          {  
            "name": "power",  
            "value": {  
              "propertyId": "a1b2c3d4-5678-90ab-cdef-66666EXAMPLE",  
              "hierarchyId": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE"  
            }  
          }  
        ],  
      },  
      "window": {  
        "tumbling": {  
          "interval": "1h"  
        }  
      }  
    }  
  }  
}
```



```

        "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"
}
}

```

Para obter mais informações, consulte [Descrição de um ativo específico](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [DescribeAsset](#) na Referência de AWS CLI Comandos.

describe-dashboard

O código de exemplo a seguir mostra como usar `describe-dashboard`.

AWS CLI

Para descrever um painel

O `describe-dashboard` exemplo a seguir descreve o painel do parque eólico especificado.

```

aws iotsitewise describe-dashboard \
  --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE

```

Saída:

```

{
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
  "dashboardName": "Wind Farm",

```

```

    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",
    "dashboardDefinition": "{\\"widgets\\":[{\\"type\\":\\"monitor-line-chart\\",\\"title
\\":\\"Generated Power\\",\\"x\\":0,\\"y\\":0,\\"height\\":3,\\"width\\":3,\\"metrics\\":
[{\\"label\\":\\"Power\\",\\"type\\":\\"iotsitewise\\",\\"assetId\\":\\"a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE\\",\\"propertyId\\":\\"a1b2c3d4-5678-90ab-cdef-99999EXAMPLE\\"}]]}]",
    "dashboardCreationDate": "2020-05-01T20:32:12.228476348Z",
    "dashboardLastUpdateDate": "2020-05-01T20:32:12.228476348Z"
}

```

Para obter mais informações, consulte [Visualização de painéis no Guia](#) de aplicativos do AWS SiteWise IoT Monitor.

- Para API obter detalhes, consulte [DescribeDashboard](#) na Referência de AWS CLI Comandos.

describe-gateway-capability-configuration

O código de exemplo a seguir mostra como usar `describe-gateway-capability-configuration`.

AWS CLI

Para descrever uma capacidade de gateway

O `describe-gateway-capability-configuration` exemplo a seguir descreve um recurso de origem OPC-UA.

```

aws iotsitewise describe-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1"

```

Saída:

```

{
  "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:green-grass-factory1-auth-3QNDmM\\"},\\"nodeFilterRules\\":
[]},\\"measurementDataStreamPrefix\\":\\"\\"}]]}",

```

```
"capabilitySyncStatus": "IN_SYNC"
}
```

Para obter mais informações, consulte [Como configurar fontes de dados](#) no Guia do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DescribeGatewayCapabilityConfiguration](#) na Referência de AWS CLI Comandos.

describe-gateway

O código de exemplo a seguir mostra como usar `describe-gateway`.

AWS CLI

Para descrever um gateway

O `describe-gateway` exemplo a seguir descreve um gateway.

```
aws iotsitewise describe-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte Como [ingerir dados usando um gateway no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DescribeGateway](#) na Referência de AWS CLI Comandos.

describe-logging-options

O código de exemplo a seguir mostra como usar `describe-logging-options`.

AWS CLI

Para recuperar as opções atuais de registro de AWS SiteWise IoT

O `describe-logging-options` exemplo a seguir recupera as opções atuais de registro de AWS SiteWise IoT para AWS sua conta na região atual.

```
aws iotsitewise describe-logging-options
```

Saída:

```
{
  "loggingOptions": {
    "level": "INFO"
  }
}
```

Para obter mais informações, consulte [Monitoramento da AWS IoT com SiteWise Amazon CloudWatch Logs no Guia](#) do usuário da AWS SiteWise IoT.

- Para API obter detalhes, consulte [DescribeLoggingOptions](#) na Referência de AWS CLI Comandos.

describe-portal

O código de exemplo a seguir mostra como usar `describe-portal`.

AWS CLI

Para descrever um portal

O `describe-portal` exemplo a seguir descreve um portal da web para uma empresa de parques eólicos.

```
aws iotsitewise describe-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Administrando seus portais no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DescribePortal](#) na Referência de AWS CLI Comandos.

describe-project

O código de exemplo a seguir mostra como usar `describe-project`.

AWS CLI

Para descrever um projeto

O `describe-project` exemplo a seguir descreve um projeto de parque eólico.

```
aws iotsitewise describe-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Visualização dos detalhes do projeto](#) no Guia de Aplicação do SiteWise Monitor de AWS IoT.

- Para API obter detalhes, consulte [DescribeProject](#) na Referência de AWS CLI Comandos.

disassociate-assets

O código de exemplo a seguir mostra como usar `disassociate-assets`.

AWS CLI

Para desassociar um ativo secundário de um ativo principal

O `disassociate-assets` exemplo a seguir dissocia um ativo de turbina eólica de um ativo de parque eólico.

```
aws iotsitewise disassociate-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-4444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-7777EXAMPLE \  
  --child-asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Associando ativos no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [DisassociateAssets](#) na Referência de AWS CLI Comandos.

get-asset-property-aggregates

O código de exemplo a seguir mostra como usar `get-asset-property-aggregates`.

AWS CLI

Para recuperar a média agregada e os valores de contagem de uma propriedade do ativo

O `get-asset-property-aggregates` exemplo a seguir recupera a potência total média de um ativo de turbina eólica e a contagem de pontos de dados de energia total por um período de 1 hora.

```
aws iotsitewise get-asset-property-aggregates \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE \  
  --start-date 1580849400 \  
  --end-date 1580853000 \  
  --aggregate-types AVERAGE COUNT \  
  --resolution 1h
```

Saída:

```
{  
  "aggregatedValues": [  
    {  
      "timestamp": 1580850000.0,  
      "quality": "GOOD",  
      "value": {  
        "average": 8723.46538886233,  
        "count": 12.0  
      }  
    }  
  ]  
}
```

Para obter mais informações, consulte [Consultar agregados de propriedades de ativos no Guia do usuário de AWS SiteWise IoT](#).

- Para API obter detalhes, consulte [GetAssetPropertyAggregates](#) na Referência de AWS CLI Comandos.

get-asset-property-value-history

O código de exemplo a seguir mostra como usar `get-asset-property-value-history`.

AWS CLI

Para recuperar os valores históricos de uma propriedade do ativo

O `get-asset-property-value-history` exemplo a seguir recupera os valores de potência total de um ativo de turbina eólica por um período de 20 minutos.

```
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
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Consultar valores históricos de propriedades de ativos](#) no Guia do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [GetAssetPropertyValueHistory](#) na Referência de AWS CLI Comandos.

get-asset-property-value

O código de exemplo a seguir mostra como usar `get-asset-property-value`.

AWS CLI

Para recuperar o valor atual de uma propriedade do ativo

O `get-asset-property-value` exemplo a seguir recupera a potência total atual de um ativo de turbina eólica.

```

aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE

```

Saída:

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

Para obter mais informações, consulte [Consultando os valores atuais das propriedades do ativo](#) no Guia do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [GetAssetPropertyValue](#) na Referência de AWS CLI Comandos.

list-access-policies

O código de exemplo a seguir mostra como usar `list-access-policies`.

AWS CLI

Para listar todas as políticas de acesso

O `list-access-policies` exemplo a seguir lista todas as políticas de acesso para um usuário que é administrador do portal.

```
aws iotsitewise list-access-policies \
  --identity-type USER \
  --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE
```

Saída:

```
{
  "accessPolicySummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",
      "identity": {
```

```
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE"
        },
        "resource": {
            "portal": {
                "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
            }
        },
        "permission": "ADMINISTRATOR"
    }
]
}
```

Para obter mais informações, consulte [Administrando seus portais no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [ListAccessPolicies](#) na Referência de AWS CLI Comandos.

list-asset-models

O código de exemplo a seguir mostra como usar `list-asset-models`.

AWS CLI

Para listar todos os modelos de ativos

O `list-asset-models` exemplo a seguir lista todos os modelos de ativos definidos em sua AWS conta na região atual.

```
aws iotsitewise list-asset-models
```

Saída:

```
{
  "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"
        }
    }
]
}

```

Para obter mais informações, consulte [Listar todos os modelos de ativos](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [ListAssetModels](#) na Referência de AWS CLI Comandos.

list-assets

O código de exemplo a seguir mostra como usar `list-assets`.

AWS CLI

Exemplo 1: Para listar todos os ativos de nível superior

O `list-assets` exemplo a seguir lista todos os ativos de nível superior na árvore hierárquica de ativos e definidos em sua AWS conta na região atual.

```

aws iotsitewise list-assets \
  --filter TOP_LEVEL

```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Listar ativos](#) no Guia do SiteWise usuário de AWS IoT.

Exemplo 2: Para listar todos os ativos com base em um modelo de ativo

O `list-assets` exemplo a seguir lista todos os ativos com base em um modelo de ativos e definidos em sua AWS conta na região atual.

```
aws iotsitewise list-assets \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Saída:

```
{
  "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": []
    }
]
}

```

Para obter mais informações, consulte [Listar ativos](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [ListAssets](#) na Referência de AWS CLI Comandos.

list-associated-assets

O código de exemplo a seguir mostra como usar `list-associated-assets`.

AWS CLI

Para listar todos os ativos associados a um ativo em uma hierarquia específica

O `list-associated-assets` exemplo a seguir lista todos os ativos da turbina eólica associados ao ativo do parque eólico especificado.

```

aws iotsitewise list-associated-assets \
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE

```

Saída:

```

{
  "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": []
    }
]
```

Para obter mais informações, consulte [Listar ativos associados a um ativo específico](#) no Guia do SiteWise Usuário de AWS IoT.

- Para API obter detalhes, consulte [ListAssociatedAssets](#) na Referência de AWS CLI Comandos.

list-dashboards

O código de exemplo a seguir mostra como usar `list-dashboards`.

AWS CLI

Para listar todos os painéis em um projeto

O `list-dashboards` exemplo a seguir lista todos os painéis definidos em um projeto.

```
aws iotsitewise list-dashboards \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Saída:

```
{
  "dashboardSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE",
      "name": "Wind Farm",
      "creationDate": "2020-05-01T20:32:12.228476348Z",
      "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"
    }
  ]
}
```

Para obter mais informações, consulte [Visualização de painéis no Guia](#) de aplicativos do AWS SiteWise IoT Monitor.

- Para API obter detalhes, consulte [ListDashboards](#) na Referência de AWS CLI Comandos.

list-gateways

O código de exemplo a seguir mostra como usar `list-gateways`.

AWS CLI

Para listar todos os gateways

O `list-gateways` exemplo a seguir lista todos os gateways definidos em sua AWS conta na região atual.

```
aws iotsitewise list-gateways
```

Saída:

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

Para obter mais informações, consulte Como [ingerir dados usando um gateway no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [ListGateways](#) na Referência de AWS CLI Comandos.

list-portals

O código de exemplo a seguir mostra como usar `list-portals`.

AWS CLI

Para listar todos os portais

O `list-portals` exemplo a seguir lista todos os portais definidos em sua AWS conta na região atual.

```
aws iotsitewise list-portals
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Administrando seus portais no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [ListPortals](#) na Referência de AWS CLI Comandos.

`list-project-assets`

O código de exemplo a seguir mostra como usar `list-project-assets`.

AWS CLI

Para listar todos os ativos associados a um projeto

O `list-project-assets` exemplo a seguir lista todos os ativos associados a um projeto de parque eólico.

```
aws iotsitewise list-projects \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Saída:

```
{  
  "assetIds": [  
    "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"  
  ]  
}
```

Para obter mais informações, consulte [Adicionar ativos a projetos](#) no Guia de aplicativos do AWS IoT SiteWise Monitor.

- Para API obter detalhes, consulte [ListProjectAssets](#) na Referência de AWS CLI Comandos.

list-projects

O código de exemplo a seguir mostra como usar `list-projects`.

AWS CLI

Para listar todos os projetos em um portal

O `list-projects` exemplo a seguir lista todos os projetos definidos em um portal.

```
aws iotsitewise list-projects \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE
```

Saída:

```
{  
  "projectSummaries": [  
    {  
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",  
      "name": "Wind Farm 1",  
      "description": "Contains asset visualizations for Wind Farm #1 for  
Example Corp.",  
      "creationDate": "2020-02-20T21:58:43.362246001Z",  
      "lastUpdateDate": "2020-02-20T21:58:43.362246095Z"  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [Visualização dos detalhes do projeto](#) no Guia de Aplicação do SiteWise Monitor de AWS IoT.

- Para API obter detalhes, consulte [ListProjects](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar todas as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um ativo de turbina eólica.

```
aws iotsitewise list-tags-for-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Saída:

```
{  
  "tags": {  
    "Owner": "richard-roe"  
  }  
}
```

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-logging-options

O código de exemplo a seguir mostra como usar `put-logging-options`.

AWS CLI

Para especificar o nível de registro

O `put-logging-options` exemplo a seguir permite o registro INFO de níveis na AWS IoT SiteWise. Outros níveis incluem DEBUG OFF e.

```
aws iotsitewise put-logging-options \  
  --logging-options level=INFO
```

Este comando não produz saída.

Para obter mais informações, consulte [Monitoramento da AWS IoT com SiteWise Amazon CloudWatch Logs no Guia](#) do usuário da AWS SiteWise IoT.

- Para API obter detalhes, consulte [PutLoggingOptions](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a um recurso

O `tag-resource` exemplo a seguir adiciona uma etiqueta de proprietário a um ativo de turbina eólica. Isso permite que você controle o acesso ao ativo com base em quem o possui.

```
aws iotsitewise tag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --tags Owner=richard-roe
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove uma etiqueta de proprietário de um ativo de turbina eólica.

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --tag-keys Owner
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-access-policy

O código de exemplo a seguir mostra como usar `update-access-policy`.

AWS CLI

Para conceder a um visualizador do projeto a propriedade de um projeto

O `update-access-policy` exemplo a seguir atualiza uma política de acesso que concede ao visualizador do projeto a propriedade de um projeto.

```
aws iotsitewise update-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \  
  --cli-input-json file://update-project-viewer-access-policy.json
```

Conteúdo de `update-project-viewer-access-policy.json`:

```
{  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
    }  
  }  
}
```

```

    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
      "project": {
        "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"
      }
    }
  }
}

```

Este comando não produz saída.

Para obter mais informações, consulte [Atribuição de proprietários de projetos](#) no Guia de aplicativos do AWS SiteWise IoT Monitor.

- Para API obter detalhes, consulte [UpdateAccessPolicy](#) na Referência de AWS CLI Comandos.

update-asset-model

O código de exemplo a seguir mostra como usar `update-asset-model`.

AWS CLI

Para atualizar um modelo de ativo

O `update-asset-model` exemplo a seguir atualiza a descrição do modelo de ativos de um parque eólico. Esse exemplo inclui o modelo existente IDs e as definições, pois `update-asset-model` substitui o modelo existente pelo novo modelo.

```

aws iotsitewise update-asset-model \
  --cli-input-json file://update-wind-farm-model.json

```

Conteúdo de `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"
    }
]
}

```

Saída:

```
{
```



```

    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelStatus": {
      "state": "CREATING"
    }
  }
}

```

Para obter mais informações, consulte [Atualização de modelos de ativos](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [UpdateAssetModel](#) na Referência de AWS CLI Comandos.

update-asset-property

O código de exemplo a seguir mostra como usar update-asset-property.

AWS CLI

Exemplo 1: Para atualizar o alias de uma propriedade do ativo

O update-asset-property exemplo a seguir atualiza o alias da propriedade de energia de um ativo de turbina eólica.

```

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

```

Este comando não produz saída.

Para obter mais informações, consulte [Mapeando fluxos de dados industriais para propriedades de ativos](#) no Guia do usuário de AWS SiteWise IoT.

Exemplo 2: Para habilitar notificações de propriedades de ativos

O update-asset-property exemplo a seguir permite notificações de atualização de propriedades de ativos para a propriedade de energia de um ativo de turbina eólica. As atualizações do valor da propriedade são publicadas no MQTT tópico \$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>, onde cada ID é substituída pela propriedade, ativo e ID do modelo da propriedade do ativo.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Interagindo com outros serviços no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [UpdateAssetProperty](#) na Referência de AWS CLI Comandos.

update-asset

O código de exemplo a seguir mostra como usar update-asset.

AWS CLI

Para atualizar o nome de um ativo

O update-asset exemplo a seguir atualiza o nome de um ativo de turbina eólica.

```
aws iotsitewise update-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --asset-name "Wind Turbine 2"
```

Saída:

```
{  
  "assetStatus": {  
    "state": "UPDATING"  
  }  
}
```

Para obter mais informações, consulte [Atualização de ativos](#) no Guia do SiteWise usuário de AWS IoT.

- Para API obter detalhes, consulte [UpdateAsset](#) na Referência de AWS CLI Comandos.

update-dashboard

O código de exemplo a seguir mostra como usar `update-dashboard`.

AWS CLI

Para atualizar um painel

O `update-dashboard` exemplo a seguir altera o título do gráfico de linhas de um painel que exibe a potência total gerada para um parque eólico.

```
aws iotsitewise update-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE \  
  --dashboard-name "Wind Farm" \  
  --dashboard-definition file://update-wind-farm-dashboard.json
```

Conteúdo de `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"  
        }  
      ]  
    }  
  ]  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de painéis \(CLI\) no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [UpdateDashboard](#) na Referência de AWS CLI Comandos.

update-gateway-capability-configuration

O código de exemplo a seguir mostra como usar `update-gateway-capability-configuration`.

AWS CLI

Para atualizar um recurso de gateway

O `update-gateway-capability-configuration` exemplo a seguir configura uma fonte OPC-UA com as seguintes propriedades:

Confia em qualquer certificado. Usa o algoritmo Basic256 para proteger mensagens. Usa o SignAndEncrypt modo para proteger conexões. Usa credenciais de autenticação armazenadas em um segredo do 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
```

Conteúdo de `opc-ua-capability-configuration.json`:

```
{
  "sources": [
    {
      "name": "Wind Farm #1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.0:49320",
        "securityPolicy": "BASIC256",
        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Username",
          "usernameSecretArn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:green-grass-windfarm1-auth-1ABCDE"
        }
      }
    }
  ]
}
```

```
        "nodeFilterRules": []
      },
      "measurementDataStreamPrefix": ""
    }
  ]
}
```

Saída:

```
{
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilitySyncStatus": "OUT_OF_SYNC"
}
```

Para obter mais informações, consulte [Como configurar fontes de dados](#) no Guia do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [UpdateGatewayCapabilityConfiguration](#) na Referência de AWS CLI Comandos.

update-gateway

O código de exemplo a seguir mostra como usar update-gateway.

AWS CLI

Para atualizar o nome de um gateway

O update-gateway exemplo a seguir atualiza o nome de um gateway.

```
aws iotsitewise update-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --gateway-name ExampleCorpGateway1
```

Este comando não produz saída.

Para obter mais informações, consulte Como [ingerir dados usando um gateway no Guia](#) do usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [UpdateGateway](#) na Referência de AWS CLI Comandos.

update-portal

O código de exemplo a seguir mostra como usar `update-portal`.

AWS CLI

Para atualizar os detalhes de um portal

O `update-portal` exemplo a seguir atualiza um portal da web para uma empresa de parques eólicos.

```
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
```

Saída:

```
{  
  "portalStatus": {  
    "state": "UPDATING"  
  }  
}
```

Para obter mais informações, consulte [Administrando seus portais no Guia](#) do Usuário de AWS SiteWise IoT.

- Para API obter detalhes, consulte [UpdatePortal](#) na Referência de AWS CLI Comandos.

update-project

O código de exemplo a seguir mostra como usar `update-project`.

AWS CLI

Para atualizar os detalhes de um projeto

O `update-project` exemplo a seguir atualiza um projeto de parque eólico.

```
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."
```

Este comando não produz saída.

Para obter mais informações, consulte [Alterando os detalhes do projeto](#) no Guia de Aplicação do SiteWise Monitor de AWS IoT.

- Para API obter detalhes, consulte [UpdateProject](#) na Referência de AWS CLI Comandos.

AWS IoT Things Graph exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Things Graph.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-entity-to-thing

O código de exemplo a seguir mostra como usar `associate-entity-to-thing`.

AWS CLI

Para associar algo a um dispositivo

O `associate-entity-to-thing` exemplo a seguir associa uma coisa a um dispositivo. O exemplo usa um dispositivo de sensor de movimento que está no namespace público.

```
aws iotthingsgraph associate-entity-to-thing \  
  --thing-name "MotionSensorName" \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e upload de modelos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [AssociateEntityToThing](#) na Referência de AWS CLI Comandos.

create-flow-template

O código de exemplo a seguir mostra como usar create-flow-template.

AWS CLI

Para criar um fluxo

O create-flow-template exemplo a seguir cria um fluxo (fluxo de trabalho). O valor de MyFlowDefinition é o GraphQL que modela o fluxo.

```
aws iotthingsgraph create-flow-template \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

Saída:

```
{  
  "summary": {  
    "createdAt": 1559248067.545,  
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",  
    "revisionNumber": 1  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [CreateFlowTemplate](#) na Referência de AWS CLI Comandos.

create-system-instance

O código de exemplo a seguir mostra como usar `create-system-instance`.

AWS CLI

Para criar uma instância do sistema

O `create-system-instance` exemplo a seguir cria uma instância do sistema. O valor de `MySystemInstanceDefinition` é o GraphQL que modela a instância do sistema.

```
aws iotthingsgraph create-system-instance -\
  -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
  --target CLOUD \
  --flow-actions-role-arn myRoleARN
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [CreateSystemInstance](#) na Referência de AWS CLI Comandos.

create-system-template

O código de exemplo a seguir mostra como usar `create-system-template`.

AWS CLI

Para criar um sistema

O `create-system-template` exemplo a seguir cria um sistema. O valor de `MySystemDefinition` é o GraphQL que modela o sistema.

```
aws iotthingsgraph create-system-template \  
  --definition language=GRAPHQL,text="MySystemDefinition"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Criação de sistemas](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [CreateSystemTemplate](#) na Referência de AWS CLI Comandos.

delete-flow-template

O código de exemplo a seguir mostra como usar `delete-flow-template`.

AWS CLI

Para excluir um fluxo

O `delete-flow-template` exemplo a seguir exclui um fluxo (fluxo de trabalho).

```
aws iotthingsgraph delete-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário do IoT Things AWS Graph](#).

- Para API obter detalhes, consulte [DeleteFlowTemplate](#) na Referência de AWS CLI Comandos.

delete-namespace

O código de exemplo a seguir mostra como usar delete-namespace.

AWS CLI

Para excluir um namespace

O delete-namespace exemplo a seguir exclui um namespace.

```
aws iotthingsgraph delete-namespace
```

Saída:

```
{
  "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
  "namespaceName": "us-west-2/123456789012/default"
}
```

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário](#) do IoT Things AWS Graph.

- Para API obter detalhes, consulte [DeleteNamespacena](#) Referência de AWS CLI Comandos.

delete-system-instance

O código de exemplo a seguir mostra como usar delete-system-instance.

AWS CLI

Para excluir uma instância do sistema

O delete-system-instance exemplo a seguir exclui uma instância do sistema.

```
aws iotthingsgraph delete-system-instance \
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário](#) do IoT Things AWS Graph.

- Para API obter detalhes, consulte [DeleteSystemInstance](#) na Referência de AWS CLI Comandos.

delete-system-template

O código de exemplo a seguir mostra como usar delete-system-template.

AWS CLI

Para excluir um sistema

O delete-system-template exemplo a seguir exclui um sistema.

```
aws iotthingsgraph delete-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário](#) do IoT Things AWS Graph.

- Para API obter detalhes, consulte [DeleteSystemTemplate](#) na Referência de AWS CLI Comandos.

deploy-system-instance

O código de exemplo a seguir mostra como usar deploy-system-instance.

AWS CLI

Para implantar uma instância do sistema

O delete-system-template exemplo a seguir implanta uma instância do sistema.

```
aws iotthingsgraph deploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

```
--id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [DeploySystemInstance](#) na Referência de AWS CLI Comandos.

deprecate-flow-template

O código de exemplo a seguir mostra como usar `deprecate-flow-template`.

AWS CLI

Para descontinuar um fluxo

O `deprecate-flow-template` exemplo a seguir desaprova um fluxo (fluxo de trabalho).

```
aws iotthingsgraph deprecate-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário](#) do IoT Things AWS Graph.

- Para API obter detalhes, consulte [DeprecateFlowTemplate](#) na Referência de AWS CLI Comandos.

deprecate-system-template

O código de exemplo a seguir mostra como usar `deprecate-system-template`.

AWS CLI

Para descontinuar um sistema

O `deprecate-system-template` exemplo a seguir desaprova um sistema.

```
aws iotthingsgraph deprecate-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário](#) do IoT Things AWS Graph.

- Para API obter detalhes, consulte [DeprecateSystemTemplate](#) na Referência de AWS CLI Comandos.

describe-namespace

O código de exemplo a seguir mostra como usar `describe-namespace`.

AWS CLI

Para obter uma descrição do seu namespace

O `describe-namespace` exemplo a seguir obtém uma descrição do seu namespace.

```
aws iotthingsgraph describe-namespace
```

Saída:

```
{  
  "namespaceName": "us-west-2/123456789012/default",  
  "trackingNamespaceName": "aws",  
  "trackingNamespaceVersion": 1,  
  "namespaceVersion": 5  
}
```

Para obter mais informações, consulte [Namespaces](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [DescribeNamespace](#) na Referência de AWS CLI Comandos.

dissociate-entity-from-thing

O código de exemplo a seguir mostra como usar `dissociate-entity-from-thing`.

AWS CLI

Para dissociar uma coisa de um dispositivo

O `dissociate-entity-from-thing` exemplo a seguir dissocia uma coisa de um dispositivo.

```
aws iotthingsgraph dissociate-entity-from-thing \  
  --thing-name "MotionSensorName" \  
  --entity-type "DEVICE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação e upload de modelos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [DissociateEntityFromThing](#) na Referência de AWS CLI Comandos.

get-entities

O código de exemplo a seguir mostra como usar `get-entities`.

AWS CLI

Para obter definições para entidades

O `get-entities` exemplo a seguir obtém uma definição para um modelo de dispositivo.

```
aws iotthingsgraph get-entities \  
  --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

Saída:

```
{
  "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}"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Criação e upload de modelos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetEntities](#) na Referência de AWS CLI Comandos.

get-flow-template-revisions

O código de exemplo a seguir mostra como usar `get-flow-template-revisions`.

AWS CLI

Para obter informações de revisão sobre um fluxo

O `get-flow-template-revisions` exemplo a seguir obtém informações de revisão sobre um fluxo (fluxo de trabalho).

```
aws iotthingsgraph get-flow-template-revisions \
  --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow
```

Saída:

```
{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
```



```

        "revisionNumber": 1,
        "createdAt": 1559247540.292
    }
]
}

```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetFlowTemplateRevisions](#) na Referência de AWS CLI Comandos.

get-flow-template

O código de exemplo a seguir mostra como usar `get-flow-template`.

AWS CLI

Para obter uma definição de fluxo

O `get-flow-template` exemplo a seguir obtém uma definição para um fluxo (fluxo de trabalho).

```

aws iotthingsgraph get-flow-template \
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"

```

Saída:

```

{
  "description": {
    "summary": {
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
      "revisionNumber": 1,
      "createdAt": 1559247540.292
    },
    "definition": {
      "language": "GRAPHQL",
      "text": "{\nquery MyFlow($camera: string!, $screen: string!)
@workflowType(id: \"urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow\")
@annotation(type: \"tgc:FlowEvent\", id: \"sledged790c1b2bcd949e09da0c9bfc077f79d
\", x: 1586, y: 653) @triggers(definition: \"{MotionSensor(description:
\\\\\"\\\\\") @position(x: 1045, y: 635.6666564941406) {\n  condition(expr:

```

```

    \\\"devices[name == \\\"\\\\\\\\\\\\\\\\\"motionSensor\\\\\\\\\\\\\\\\\"].events[name == \\\"
    \\\"StateChanged\\\\\\\\\\\\\\\\\"].lastEvent\\\\\\\"))\\n  action(expr: \\\"\\\\\\\\\\\\\\\\\")\\
    \\n}}\\\")) {\\n  variables {\\n    cameraResult @property(id: \\\"urn:tdm:aws/
    examples:property:CameraStateProperty\\\")\\n  }\\n  steps {\\n    step(name: \\\"Camera
    \\\", outEvent: [\\\"sledged790c1b2bcd949e09da0c9bfc077f79d\\\"] @position(x: 1377,
    y: 638.6666564941406) {\\n      DeviceActivity(deviceModel: \\\"urn:tdm:aws/
    examples:deviceModel:Camera\\\", out: \\\"cameraResult\\\", deviceId: \\\"${camera}\\\")
    {\\n        capture\\n      }\\n    }\\n    step(name: \\\"Screen\\\", inEvent:
    [\\\"sledged790c1b2bcd949e09da0c9bfc077f79d\\\"] @position(x: 1675.6666870117188,
    y: 637.9999847412109) {\\n      DeviceActivity(deviceModel: \\\"urn:tdm:aws/
    examples:deviceModel:Screen\\\", deviceId: \\\"${screen}\\\") {\\n        display(imageUrl:
    \\\"${cameraResult.lastClickedImage}\\\")\\n      }\\n    }\\n  }\\n}\\n}\\n}
  },
  \"validatedNamespaceVersion\": 5
}
}

```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetFlowTemplate](#) na Referência de AWS CLI Comandos.

get-namespace-deletion-status

O código de exemplo a seguir mostra como usar get-namespace-deletion-status.

AWS CLI

Para obter o status da tarefa de exclusão do namespace

O get-namespace-deletion-status exemplo a seguir obtém o status da tarefa de exclusão do namespace.

```
aws iotthingsgraph get-namespace-deletion-status
```

Saída:

```

{
  \"namespaceArn\": \"arn:aws:iotthingsgraph:us-west-2:123456789012\",
  \"namespaceName\": \"us-west-2/123456789012/default\"
  \"status\": \"SUCCEEDED \"
}

```

Para obter mais informações, consulte [Namespaces](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetNamespaceDeletionStatus](#) na Referência de AWS CLI Comandos.

get-system-instance

O código de exemplo a seguir mostra como usar `get-system-instance`.

AWS CLI

Para obter uma instância do sistema

O `get-system-instance` exemplo a seguir obtém uma definição para uma instância do sistema.

```
aws iotthingsgraph get-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Saída:

```
{  
  "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"
}
}

```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetSystemInstancena](#) Referência de AWS CLI Comandos.

get-system-template-revisions

O código de exemplo a seguir mostra como usar `get-system-template-revisions`.

AWS CLI

Para obter informações de revisão sobre um sistema

O `get-system-template-revisions` exemplo a seguir obtém informações de revisão sobre um sistema.

```

aws iotthingsgraph get-system-template-revisions \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"

```

Saída:

```

{
  "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
    }
  ]
}

```

```
}

```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetSystemTemplateRevisions](#) na Referência de AWS CLI Comandos.

get-system-template

O código de exemplo a seguir mostra como usar `get-system-template`.

AWS CLI

Para obter um sistema

O `get-system-template` exemplo a seguir obtém uma definição para um sistema.

```
aws iotthingsgraph get-system-template \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetSystemTemplate](#) na Referência de AWS CLI Comandos.

get-upload-status

O código de exemplo a seguir mostra como usar `get-upload-status`.

AWS CLI

Para obter o status da sua entidade, faça o upload

O `get-upload-status` exemplo a seguir obtém o status da operação de upload da sua entidade. O valor de `MyUploadId` é o valor do ID retornado pela `upload-entity-definitions` operação.

```
aws iotthingsgraph get-upload-status \
  --upload-id "MyUploadId"
```

Saída:

```
{
  "namespaceName": "us-west-2/123456789012/default",
  "namespaceVersion": 5,
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
  "uploadStatus": "SUCCEEDED"
}
```

Para obter mais informações, consulte [Modelagem de entidades](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [GetUploadStatus](#) na Referência de AWS CLI Comandos.

list-flow-execution-messages

O código de exemplo a seguir mostra como usar `list-flow-execution-messages`.

AWS CLI

Para obter informações sobre eventos em uma execução de fluxo

O `list-flow-execution-messages` exemplo a seguir obtém informações sobre eventos em uma execução de fluxo.

```
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"
```

Saída:

```
{
  "messages": [
    {
      "eventType": "EXECUTION_STARTED",
      "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
      "payload": "Flow execution started",
      "timestamp": 1559247540.656
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [ListFlowExecutionMessages](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar todas as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um recurso do AWS IoT Things Graph.

```
aws iotthingsgraph list-tags-for-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218"
```

Saída:

```
{  
  "tags": [  
    {  
      "key": "Type",  
      "value": "Residential"  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [marcar seus recursos do AWS IoT Things Graph](#) no Guia do usuário AWS do IoT Things Graph.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

search-entities

O código de exemplo a seguir mostra como usar search-entities.

AWS CLI

Para pesquisar entidades

O search-entities exemplo a seguir pesquisa todas as entidades do tipoEVENT.

```
aws iotthingsgraph search-entities \  
  --entity-types "EVENT"
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte Referência do [modelo de dados do AWS IoT Things Graph](#) no Guia AWS do usuário do IoT Things Graph.

- Para API obter detalhes, consulte [SearchEntities](#) na Referência de AWS CLI Comandos.

search-flow-executions

O código de exemplo a seguir mostra como usar search-flow-executions.

AWS CLI

Para pesquisar execuções de fluxo

O `search-flow-executions` exemplo a seguir pesquisa todas as execuções de um fluxo em uma instância do sistema especificada.

```
aws iotthingsgraph search-flow-executions \  
  --system-instance-id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [SearchFlowExecutions](#) na Referência de AWS CLI Comandos.

search-flow-templates

O código de exemplo a seguir mostra como usar `search-flow-templates`.

AWS CLI

Para pesquisar fluxos (ou fluxos de trabalho)

O `search-flow-templates` exemplo a seguir pesquisa todos os fluxos (fluxos de trabalho) que contêm o modelo do dispositivo Camera.

```
aws iotthingsgraph search-flow-templates \  
  --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [SearchFlowTemplates](#) na Referência de AWS CLI Comandos.

search-system-instances

O código de exemplo a seguir mostra como usar `search-system-instances`.

AWS CLI

Para pesquisar instâncias do sistema

O `search-system-instances` exemplo a seguir pesquisa todas as instâncias do sistema que contêm o sistema especificado.

```
aws iotthingsgraph search-system-instances \  
  --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/  
  default:System:SecurityFlow"
```

Saída:

```
{
  "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
    }
]
}

```

Para obter mais informações, consulte Como [trabalhar com sistemas e configurações de fluxo](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [SearchSystemInstances](#) na Referência de AWS CLI Comandos.

search-system-templates

O código de exemplo a seguir mostra como usar search-system-templates.

AWS CLI

Para pesquisar o sistema

O search-system-templates exemplo a seguir pesquisa todos os sistemas que contêm o fluxo especificado.

```

aws iotthingsgraph search-system-templates \
  --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow"

```

Saída:

```
{
```

```
    "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  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [SearchSystemTemplates](#) na Referência de AWS CLI Comandos.

search-things

O código de exemplo a seguir mostra como usar search-things.

AWS CLI

Para pesquisar itens associados a dispositivos e modelos de dispositivos

O search-things exemplo a seguir pesquisa todas as coisas associadas ao MotionSensor dispositivo HCSR5 01.

```
aws iotthingsgraph search-things \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [Criação e upload de modelos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [SearchThings](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para criar uma tag para um recurso

O `tag-resource` exemplo a seguir cria uma tag para o recurso especificado.

```
aws iotthingsgraph tag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218" \  
  --tags key="Type",value="Residential"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos do AWS IoT Things Graph](#) no Guia do usuário AWS do IoT Things Graph.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

undeploy-system-instance

O código de exemplo a seguir mostra como usar `undeploy-system-instance`.

AWS CLI

Para desimplantar uma instância do sistema de seu destino

O `undeploy-system-instance` exemplo a seguir remove uma instância do sistema de seu destino.

```
aws iotthingsgraph undeploy-system-instance \  
  --target-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218"
```

```
--id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Gerenciamento do ciclo de vida para entidades, fluxos, sistemas e implantações do AWS IoT Things Graph no Guia do usuário](#) do IoT Things AWS Graph.

- Para API obter detalhes, consulte [UndeploySystemInstance](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove uma tag para o recurso especificado.

```
aws iotthingsgraph untag-resource \
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/default/Room218" \
  --tag-keys "Type"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus recursos do AWS IoT Things Graph](#) no Guia do usuário AWS do IoT Things Graph.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-flow-template

O código de exemplo a seguir mostra como usar `update-flow-template`.

AWS CLI

Para atualizar um fluxo

O `update-flow-template` exemplo a seguir atualiza um fluxo (fluxo de trabalho). O valor de `MyFlowDefinition` é o GraphQL que modela o fluxo.

```
aws iotthingsgraph update-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

Saída:

```
{  
  "summary": {  
    "createdAt": 1559248067.545,  
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",  
    "revisionNumber": 2  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com fluxos no Guia](#) do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [UpdateFlowTemplate](#) na Referência de AWS CLI Comandos.

update-system-template

O código de exemplo a seguir mostra como usar `update-system-template`.

AWS CLI

Para atualizar um sistema

O `update-system-template` exemplo a seguir atualiza um sistema. O valor de `MySystemDefinition` é o GraphQL que modela o sistema.

```
aws iotthingsgraph update-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \  
  --definition language=GRAPHQL,text="MySystemDefinition"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Criação de sistemas](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [UpdateSystemTemplate](#) na Referência de AWS CLI Comandos.

upload-entity-definitions

O código de exemplo a seguir mostra como usar `upload-entity-definitions`.

AWS CLI

Para carregar definições de entidade

O `upload-entity-definitions` exemplo a seguir carrega as definições de entidade no seu namespace. O valor de `MyEntityDefinitions` é o GraphQL que modela as entidades.

```
aws iotthingsgraph upload-entity-definitions \  
  --document language=GRAPHQL,text="MyEntityDefinitions"
```

Saída:

```
{  
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"  
}
```

Para obter mais informações, consulte [Modelagem de entidades](#) no Guia do usuário do AWS IoT Things Graph.

- Para API obter detalhes, consulte [UploadEntityDefinitions](#) na Referência de AWS CLI Comandos.

AWS IoT Wireless exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS IoT Wireless.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-aws-account-with-partner-account

O código de exemplo a seguir mostra como usar `associate-aws-account-with-partner-account`.

AWS CLI

Para associar uma conta de parceiro à sua AWS conta

O `associate-aws-account-with-partner-account` exemplo a seguir associa as seguintes credenciais da conta Sidewalk à sua conta. AWS

```
aws iotwireless associate-aws-account-with-partner-account \
```

```
--sidewalk
AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78
```

Saída:

```
{
  "Sidewalk": {
    "AmazonId": "12345678901234",
    "AppServerPrivateKey":
"a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
  }
}
```

Para obter mais informações, consulte [Integração do Amazon Sidewalk para AWS IoT Core no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [AssociateAwsAccountWithPartnerAccount](#) na Referência de AWS CLI Comandos.

associate-wireless-device-with-thing

O código de exemplo a seguir mostra como usar `associate-wireless-device-with-thing`.

AWS CLI

Para associar algo a um dispositivo sem fio

O `associate-wireless-device-with-thing` exemplo a seguir associa algo ao seu dispositivo sem fio que tem o ID especificado.

```
aws iotwireless associate-wireless-device-with-thing \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [AssociateWirelessDeviceWithThing](#) na Referência de AWS CLI Comandos.

associate-wireless-gateway-with-certificate

O código de exemplo a seguir mostra como usar `associate-wireless-gateway-with-certificate`.

AWS CLI

Para associar o certificado ao gateway sem fio

O seguinte `associate-wireless-gateway-with-certificate` associa um gateway sem fio a um certificado.

```
aws iotwireless associate-wireless-gateway-with-certificate \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --iot-certificate-
id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

Saída:

```
{
  "IotCertificateId":
  "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
}
```

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [AssociateWirelessGatewayWithCertificate](#) na Referência de AWS CLI Comandos.

associate-wireless-gateway-with-thing

O código de exemplo a seguir mostra como usar `associate-wireless-gateway-with-thing`.

AWS CLI

Para associar algo a um gateway sem fio

O `associate-wireless-gateway-with-thing` exemplo a seguir associa algo a um gateway sem fio.

```
aws iotwireless associate-wireless-gateway-with-thing \
```

```
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
--thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [AssociateWirelessGatewayWithThing](#) na Referência de AWS CLI Comandos.

create-destination

O código de exemplo a seguir mostra como usar create-destination.

AWS CLI

Para criar um destino sem fio de IoT

O create-destination exemplo a seguir cria um destino para mapear uma mensagem do dispositivo para uma regra de AWS IoT. Antes de executar esse comando, você deve ter criado uma IAM função que forneça ao AWS IoT Core LoRa WAN as permissões necessárias para enviar dados para a regra de IoT AWS .

```
aws iotwireless create-destination \  
  --name IoTWirelessDestination \  
  --expression-type RuleName \  
  --expression IoTWirelessRule \  
  --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

Saída:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination",  
  "Name": "IoTWirelessDestination"  
}
```

Para obter mais informações, consulte [Adicionar destinos ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [CreateDestination](#) na Referência de AWS CLI Comandos.

create-device-profile

O código de exemplo a seguir mostra como usar `create-device-profile`.

AWS CLI

Para criar um novo perfil de dispositivo

O `create-device-profile` exemplo a seguir cria um novo perfil de dispositivo sem fio de IoT.

```
aws iotwireless create-device-profile
```

Saída:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [CreateDeviceProfile](#) na Referência de AWS CLI Comandos.

create-service-profile

O código de exemplo a seguir mostra como usar `create-service-profile`.

AWS CLI

Para criar um novo perfil de serviço

O `create-service-profile` exemplo a seguir cria um novo perfil de serviço sem fio de IoT.

```
aws iotwireless create-service-profile
```

Saída:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [CreateServiceProfile](#) na Referência de AWS CLI Comandos.

create-wireless-device

O código de exemplo a seguir mostra como usar `create-wireless-device`.

AWS CLI

Para criar um dispositivo sem fio de IoT

O `create-wireless-device` exemplo a seguir cria um recurso de dispositivo sem fio do tipo LoRaWAN.

```
aws iotwireless create-wireless-device \
  --cli-input-json file://input.json
```

Conteúdo de `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
}
```

Saída:

```
{
  "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1fffd32c8-8130-4194-96df-622f072a315f",
  "Id": "1fffd32c8-8130-4194-96df-622f072a315f"
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateWirelessDevice](#) na Referência de AWS CLI Comandos.

create-wireless-gateway-task-definition

O código de exemplo a seguir mostra como usar `create-wireless-gateway-task-definition`.

AWS CLI

Para criar uma definição de tarefa de gateway sem fio

O seguinte cria tarefas `create-wireless-gateway-task-definition` automaticamente usando essa definição de tarefa para todos os gateways com a versão atual especificada.

```
aws iotwireless create-wireless-gateway-task-definition \
  --cli-input-json file://input.json
```

Conteúdo de `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"
    }
}
}
}
}

```

Saída:

```

{
  "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
}

```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateWirelessGatewayTaskDefinition](#) na Referência de AWS CLI Comandos.

create-wireless-gateway-task

O código de exemplo a seguir mostra como usar create-wireless-gateway-task.

AWS CLI

Para criar a tarefa para um gateway sem fio

O create-wireless-gateway-task exemplo a seguir cria uma tarefa para um gateway sem fio.

```

aws iotwireless create-wireless-gateway-task \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"

```

Saída:

```
{
  "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",
  "Status": "Success"
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateWirelessGatewayTask](#) na Referência de AWS CLI Comandos.

create-wireless-gateway

O código de exemplo a seguir mostra como usar `create-wireless-gateway`.

AWS CLI

Para criar um gateway sem fio

O `create-wireless-gateway` exemplo a seguir cria um gateway de LoRa WAN dispositivo sem fio.

```
aws iotwireless create-wireless-gateway \
  --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
  --name "myFirstLoRaWANGateway" \
  --description "Using my first LoRaWAN gateway"
```

Saída:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [CreateWirelessGateway](#) na Referência de AWS CLI Comandos.

delete-destination

O código de exemplo a seguir mostra como usar delete-destination.

AWS CLI

Para excluir um destino sem fio de IoT

O delete-destination exemplo a seguir exclui o recurso de destino sem fio com o nome IoWirelessDestination que você criou.

```
aws iotwireless delete-destination \  
  --name "IoWirelessDestination"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar destinos ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [DeleteDestination](#) na Referência de AWS CLI Comandos.

delete-device-profile

O código de exemplo a seguir mostra como usar delete-device-profile.

AWS CLI

Para excluir um perfil de dispositivo

O delete-device-profile exemplo a seguir exclui um perfil de dispositivo com a ID especificada que você criou.

```
aws iotwireless delete-device-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [DeleteDeviceProfile](#) na Referência de AWS CLI Comandos.

delete-service-profile

O código de exemplo a seguir mostra como usar delete-service-profile.

AWS CLI

Para excluir um perfil de serviço

O delete-service-profile exemplo a seguir exclui um perfil de serviço com a ID especificada que você criou.

```
aws iotwireless delete-service-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [DeleteServiceProfile](#) na Referência de AWS CLI Comandos.

delete-wireless-device

O código de exemplo a seguir mostra como usar delete-wireless-device.

AWS CLI

Para excluir um dispositivo sem fio

O delete-wireless-device exemplo a seguir exclui um dispositivo sem fio que tem a ID especificada.

```
aws iotwireless delete-wireless-device \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core no Guia do LoRa WAN Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [DeleteWirelessDevice](#) na Referência de AWS CLI Comandos.

delete-wireless-gateway-task-definition

O código de exemplo a seguir mostra como usar `delete-wireless-gateway-task-definition`.

AWS CLI

Para excluir uma definição de tarefa de gateway sem fio

O `delete-wireless-gateway-task-definition` exemplo a seguir exclui a definição de tarefa do gateway sem fio que você criou com a seguinte ID.

```
aws iotwireless delete-wireless-gateway-task-definition \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteWirelessGatewayTaskDefinition](#) na Referência de AWS CLI Comandos.

delete-wireless-gateway-task

O código de exemplo a seguir mostra como usar `delete-wireless-gateway-task`.

AWS CLI

Para excluir uma tarefa de gateway sem fio

O `delete-wireless-gateway-task` exemplo a seguir exclui a tarefa do gateway sem fio que tem a ID especificada.

```
aws iotwireless delete-wireless-gateway-task \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteWirelessGatewayTask](#) na Referência de AWS CLI Comandos.

delete-wireless-gateway

O código de exemplo a seguir mostra como usar `delete-wireless-gateway`.

AWS CLI

Para excluir um gateway sem fio

O `delete-wireless-gateway` exemplo a seguir exclui um gateway sem fio que tem a ID especificada.

```
aws iotwireless delete-wireless-gateway \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DeleteWirelessGateway](#) na Referência de AWS CLI Comandos.

disassociate-aws-account-from-partner-account

O código de exemplo a seguir mostra como usar `disassociate-aws-account-from-partner-account`.

AWS CLI

Para desassociar a conta do parceiro da AWS conta

O `disassociate-aws-account-from-partner-account` exemplo a seguir desassocia uma conta de parceiro da sua conta atualmente associada AWS .

```
aws iotwireless disassociate-aws-account-from-partner-account \  
  --partner-account-id "12345678901234" \  
  --partner-type "Sidewalk"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DisassociateAwsAccountFromPartnerAccount](#) na Referência de AWS CLI Comandos.

disassociate-wireless-device-from-thing

O código de exemplo a seguir mostra como usar `disassociate-wireless-device-from-thing`.

AWS CLI

Para desassociar a coisa do dispositivo sem fio

O `disassociate-wireless-device-from-thing` exemplo a seguir dissocia um dispositivo sem fio de sua coisa atualmente associada.

```
aws iotwireless disassociate-wireless-device-from-thing \  
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DisassociateWirelessDeviceFromThing](#) na Referência de AWS CLI Comandos.

disassociate-wireless-gateway-from-certificate

O código de exemplo a seguir mostra como usar `disassociate-wireless-gateway-from-certificate`.

AWS CLI

Para desassociar o certificado do gateway sem fio

O seguinte `disassociate-wireless-gateway-from-certificate` dissocia um gateway sem fio de seu certificado atualmente associado.


```
aws iotwireless disassociate-wireless-gateway-from-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DisassociateWirelessGatewayFromCertificate](#) na Referência de AWS CLI Comandos.

disassociate-wireless-gateway-from-thing

O código de exemplo a seguir mostra como usar `disassociate-wireless-gateway-from-thing`.

AWS CLI

Para desassociar a coisa do gateway sem fio

O `disassociate-wireless-gateway-from-thing` exemplo a seguir dissocia um gateway sem fio de sua coisa atualmente associada.

```
aws iotwireless disassociate-wireless-gateway-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar seus gateways e dispositivos sem fio ao AWS IoT Core no Guia](#) do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [DisassociateWirelessGatewayFromThing](#) na Referência de AWS CLI Comandos.

get-destination

O código de exemplo a seguir mostra como usar `get-destination`.

AWS CLI

Para obter informações sobre um destino sem fio de IoT

O `get-destination` exemplo a seguir obtém informações sobre o recurso de destino com o nome `IoTWirelessDestination` que você criou.

```
aws iotwireless get-destination \  
  --name "IoTWirelessDestination"
```

Saída:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination",  
  "Name": "IoTWirelessDestination",  
  "Expression": "IoTWirelessRule",  
  "ExpressionType": "RuleName",  
  "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"  
}
```

Para obter mais informações, consulte [Adicionar destinos ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [GetDestination](#) na Referência de AWS CLI Comandos.

get-device-profile

O código de exemplo a seguir mostra como usar `get-device-profile`.

AWS CLI

Para obter informações sobre um perfil de dispositivo

O `get-device-profile` exemplo a seguir obtém informações sobre o perfil do dispositivo com a ID especificada que você criou.

```
aws iotwireless get-device-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Saída:

```
{  
  "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
}
}
```

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [GetDeviceProfile](#) na Referência de AWS CLI Comandos.

get-partner-account

O código de exemplo a seguir mostra como usar get-partner-account.

AWS CLI

Para obter as informações da conta do parceiro

O get-partner-account exemplo a seguir obtém informações sobre sua conta do Sidewalk que tem o seguinte ID.

```
aws iotwireless get-partner-account \
  --partner-account-id "12345678901234" \
  --partner-type "Sidewalk"
```

Saída:

```
{
  "Sidewalk": {
    "AmazonId": "12345678901234",
    "Fingerprint":
"a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
  },
  "AccountLinked": false
}
```



```
LnJvb3RjYTEuYW1hem9udHJ1c3QuY29tL3Jvb3RjYTEuY3JsMBMGA1UdIAQMMAow\n
CAYGZ4EMAQIBMA0GCSqGSIb3DQEBwUAA4IBAQCfkr41u3nP04FCH0TjY3NT0VI1\n
59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss2lJwJFqMp2PphKg5625kXg8kP2CN5t\n
6G7bMQcT8C8xDZntYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n
8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB480z+ctBWuZSkbvkNodPLamkB2g1\n
upRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtKPISLMVCx7i4hncxHZS\n
yLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n
-----END CERTIFICATE-----\n"
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetServiceEndpoint](#) na Referência de AWS CLI Comandos.

get-service-profile

O código de exemplo a seguir mostra como usar `get-service-profile`.

AWS CLI

Para obter informações sobre um perfil de serviço

O `get-service-profile` exemplo a seguir obtém informações sobre o perfil de serviço com a ID especificada que você criou.

```
aws iotwireless get-service-profile \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Saída:

```
{
  "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  
  }  
}
```

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [GetServiceProfile](#) na Referência de AWS CLI Comandos.

get-wireless-device-statistics

O código de exemplo a seguir mostra como usar `get-wireless-device-statistics`.

AWS CLI

Para obter informações operacionais sobre um dispositivo sem fio

O `get-wireless-device-statistics` exemplo a seguir obtém informações operacionais sobre um dispositivo sem fio.

```
aws iotwireless get-wireless-device-statistics \  
  --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

Saída:

```
{  
  "WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessDeviceStatistics](#) na Referência de AWS CLI Comandos.

get-wireless-device

O código de exemplo a seguir mostra como usar `get-wireless-device`.

AWS CLI

Para obter informações sobre o dispositivo sem fio

O `get-wireless-device` exemplo a seguir lista os widgets disponíveis na sua AWS conta.

```
aws iotwireless get-wireless-device \  
  --identifier "1fffd32c8-8130-4194-96df-622f072a315f" \  
  --identifier-type WirelessDeviceID
```

Saída:

```
{  
  "Name": "myLoRaWANDevice",  
  "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/44b87eb4-9bce-423d-  
b5fc-973f5ecc358b",  
  "DestinationName": "IoTWirelessDestination",  
  "Id": "1fffd32c8-8130-4194-96df-622f072a315f",  
  "ThingName": "44b87eb4-9bce-423d-b5fc-973f5ecc358b",  
  "Type": "LoRaWAN",  
  "LoRaWAN": {  
    "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",  
    "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",  
    "OtaaV1_1": {  
      "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",  
      "JoinEui": "b4c231a359bc2e3d",  
      "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"  
    },  
    "DevEui": "ac12efc654d23fc2"  
  },  
  "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:WirelessDevice/1fffd32c8-8130-4194-96df-622f072a315f",  
  "Description": "My LoRaWAN wireless device"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessDevice](#) na Referência de AWS CLI Comandos.

get-wireless-gateway-certificate

O código de exemplo a seguir mostra como usar `get-wireless-gateway-certificate`.

AWS CLI

Para obter a ID de um certificado associado a um gateway sem fio

O `get-wireless-gateway-certificate` exemplo a seguir obtém a ID do certificado associada a um gateway sem fio que tem a ID especificada.

```
aws iotwireless get-wireless-gateway-certificate \  
  --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

Saída:

```
{  
  "IotCertificateId":  
  "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessGatewayCertificate](#) na Referência de AWS CLI Comandos.

get-wireless-gateway-firmware-information

O código de exemplo a seguir mostra como usar `get-wireless-gateway-firmware-information`.

AWS CLI

Para obter informações de firmware sobre um gateway sem fio

O `get-wireless-gateway-firmware-information` exemplo a seguir obtém a versão do firmware e outras informações sobre um gateway sem fio.

```
aws iotwireless get-wireless-gateway-firmware-information \  
  --id "3039b406-5cc9-4307-925b-9948c63da25b"
```


Saída:

```
{
  "LoRaWAN" :{
    "CurrentVersion" :{
      "PackageVersion" : "1.0.0",
      "Station" : "2.0.5",
      "Model" : "linux"
    }
  }
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessGatewayFirmwareInformation](#) na Referência de AWS CLI Comandos.

get-wireless-gateway-statistics

O código de exemplo a seguir mostra como usar `get-wireless-gateway-statistics`.

AWS CLI

Para obter informações operacionais sobre um gateway sem fio

O `get-wireless-gateway-statistics` exemplo a seguir obtém informações operacionais sobre um gateway sem fio.

```
aws iotwireless get-wireless-gateway-statistics \
  --wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Saída:

```
{
  "WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessGatewayStatistics](#) na Referência de AWS CLI Comandos.

get-wireless-gateway-task-definition

O código de exemplo a seguir mostra como usar `get-wireless-gateway-task-definition`.

AWS CLI

Para obter informações sobre uma definição de tarefa de gateway sem fio

O `get-wireless-gateway-task-definition` exemplo a seguir obtém informações sobre a definição da tarefa sem fio com o ID especificado.

```
aws iotwireless get-wireless-gateway-task-definition \
  --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

Saída:

```
{
  "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"
      }
    }
  }
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessGatewayTaskDefinition](#) na Referência de AWS CLI Comandos.

get-wireless-gateway-task

O código de exemplo a seguir mostra como usar `get-wireless-gateway-task`.

AWS CLI

Para obter informações sobre a tarefa do gateway sem fio

O `get-wireless-gateway-task` exemplo a seguir obtém informações sobre a tarefa do gateway sem fio com a ID especificada.

```
aws iotwireless get-wireless-gateway-task \  
  --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

Saída:

```
{  
  "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",  
  "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",  
  "Status": "Success"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessGatewayTask](#) na Referência de AWS CLI Comandos.

get-wireless-gateway

O código de exemplo a seguir mostra como usar `get-wireless-gateway`.

AWS CLI

Para obter informações sobre um gateway sem fio

O `get-wireless-gateway` exemplo a seguir obtém informações sobre o gateway sem fio `myFirstLoRaWANGateway`.

```
aws iotwireless get-wireless-gateway \  
  --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --identifier-type WirelessGatewayId
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [GetWirelessGateway](#) na Referência de AWS CLI Comandos.

list-destinations

O código de exemplo a seguir mostra como usar `list-destinations`.

AWS CLI

Para listar os destinos sem fio

O `list-destinations` exemplo a seguir lista os destinos disponíveis registrados em sua AWS conta.

```
aws iotwireless list-destinations
```

Saída:

```
{
```

```

    "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Adicionar destinos ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [ListDestinations](#) na Referência de AWS CLI Comandos.

list-device-profiles

O código de exemplo a seguir mostra como usar `list-device-profiles`.

AWS CLI

Para listar os perfis do dispositivo

O `list-device-profiles` exemplo a seguir lista os perfis de dispositivos disponíveis registrados em sua AWS conta.

```
aws iotwireless list-device-profiles
```

Saída:

```

{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [ListDeviceProfiles](#) na Referência de AWS CLI Comandos.

list-partner-accounts

O código de exemplo a seguir mostra como usar `list-partner-accounts`.

AWS CLI

Para listar as contas de parceiros

O `list-partner-accounts` exemplo a seguir lista as contas de parceiros disponíveis associadas à sua AWS conta.

```
aws iotwireless list-partner-accounts
```

Saída:

```
{
  "Sidewalk": [
    {
      "AmazonId": "78965678771228",
      "Fingerprint":
"bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
    },
    {
      "AmazonId": "89656787651228",
      "Fingerprint":
"bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
    }
  ]
}
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [Integração do Amazon Sidewalk para AWS IoT Core no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [ListPartnerAccounts](#) na Referência de AWS CLI Comandos.

list-service-profiles

O código de exemplo a seguir mostra como usar `list-service-profiles`.

AWS CLI

Para listar os perfis de serviço

O `list-service-profiles` exemplo a seguir lista os perfis de serviço disponíveis registrados em sua AWS conta.

```
aws iotwireless list-service-profiles
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Adicionar perfis ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [ListServiceProfiles](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags atribuídas ao recurso

O `list-tags-for-resource` exemplo a seguir lista as tags atribuídas a um recurso de destino sem fio.

```
aws iotwireless list-tags-for-resource \
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
  IoTWirelessDestination"
```

Saída:

```
{
  "Tags": [
    {
      "Value": "MyValue",
      "Key": "MyTag"
    }
  ]
}
```

Para obter mais informações, consulte [Descreva seu AWS IoT Core para obter LoRa WAN recursos no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-wireless-devices

O código de exemplo a seguir mostra como usar `list-wireless-devices`.

AWS CLI

Para listar os dispositivos sem fio disponíveis

O `list-wireless-devices` exemplo a seguir lista os dispositivos sem fio disponíveis registrados em sua AWS conta.


```
aws iotwireless list-wireless-devices
```

Saída:

```
{
  "WirelessDeviceList": [
    {
      "Name": "myLoRaWANDevice",
      "DestinationName": "IoTWirelessDestination",
      "Id": "1fffd32c8-8130-4194-96df-622f072a315f",
      "Type": "LoRaWAN",
      "LoRaWAN": {
        "DevEui": "ac12efc654d23fc2"
      },
      "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1fffd32c8-8130-4194-96df-622f072a315f"
    }
  ]
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListWirelessDevices](#) na Referência de AWS CLI Comandos.

list-wireless-gateway-task-definitions

O código de exemplo a seguir mostra como usar `list-wireless-gateway-task-definitions`.

AWS CLI

Para listar as definições de tarefas do gateway sem fio

O `list-wireless-gateway-task-definitions` exemplo a seguir lista as definições de tarefas de gateway sem fio disponíveis registradas AWS em sua conta.

```
aws iotwireless list-wireless-gateway-task-definitions
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListWirelessGatewayTaskDefinitions](#) na Referência de AWS CLI Comandos.

list-wireless-gateways

O código de exemplo a seguir mostra como usar `list-wireless-gateways`.

AWS CLI

Para listar os gateways sem fio

O `list-wireless-gateways` exemplo a seguir lista os gateways sem fio disponíveis em sua AWS conta.

```
aws iotwireless list-wireless-gateways
```

Saída:

```
{
  "WirelessGatewayList": [
```

```

    {
      "Description": "My first LoRaWAN gateway",
      "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "dac632ebc01d23e4"
      },
      "Id": "3039b406-5cc9-4307-925b-9948c63da25b",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3039b406-5cc9-4307-925b-9948c63da25b",
      "Name": "myFirstLoRaWANGateway"
    },
    {
      "Description": "My second LoRaWAN gateway",
      "LoRaWAN": {
        "RfRegion": "US915",
        "GatewayEui": "cda123ffffe92ecd2"
      },
      "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Name": "mySecondLoRaWANGateway"
    }
  ]
}

```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [ListWirelessGateways](#) na Referência de AWS CLI Comandos.

send-data-to-wireless-device

O código de exemplo a seguir mostra como usar `send-data-to-wireless-device`.

AWS CLI

Para enviar dados para o dispositivo sem fio

O `send-data-to-wireless-device` exemplo a seguir envia um quadro de dados de aplicativo descryptografado para o dispositivo sem fio.

```

aws iotwireless send-data-to-wireless-device \
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \

```

```
--transmit-mode "1" \  
--payload-data "SGVsbG8gVG8gRGV2c2lt" \  
--wireless-metadata LoRaWAN={FPort=1}
```

Saída:

```
{  
  MessageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [SendDataToWirelessDevice](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para especificar uma tag, chave e valor para um recurso

O tag-resource exemplo a seguir marca o destino sem fio `IoTWirelessDestination` com a chave `MyTag` e o valor `MyValue`.

```
aws iotwireless tag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/  
IoTWirelessDestination" \  
  --tags Key="MyTag",Value="MyValue"
```

Este comando não produz saída.

Para obter mais informações, consulte [Descreva seu AWS IoT Core para obter LoRa WAN recursos no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

test-wireless-device

O código de exemplo a seguir mostra como usar test-wireless-device.

AWS CLI

Para testar o dispositivo sem fio

O `test-wireless-device` exemplo a seguir envia dados de uplink de Hello para um dispositivo com ID especificado.

```
aws iotwireless test-wireless-device \  
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

Saída:

```
{  
  Result: "Test succeeded. one message is sent with payload: hello"  
}
```

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [TestWirelessDevice](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma ou mais tags de um recurso

O `untag-resource` exemplo a seguir remove a tag `MyTag` e seu valor do destino sem fio `IoTWirelessDestination`.

```
aws iotwireless untag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination" \  
  --tag-keys "MyTag"
```

Este comando não produz saída.

Para obter mais informações, consulte [Descreva seu AWS IoT Core para obter LoRa WAN recursos no Guia para](#) desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-destination

O código de exemplo a seguir mostra como usar `update-destination`.

AWS CLI

Para atualizar as propriedades de um destino

O `update-destination` exemplo a seguir atualiza a propriedade de descrição de um destino sem fio.

```
aws iotwireless update-destination \  
  --name "IoTWirelessDestination" \  
  --description "Destination for messages processed using IoTWirelessRule"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar destinos ao AWS IoT Core LoRa WAN no Guia do Desenvolvedor](#) de AWS IoT.

- Para API obter detalhes, consulte [UpdateDestination](#) na Referência de AWS CLI Comandos.

update-partner-account

O código de exemplo a seguir mostra como usar `update-partner-account`.

AWS CLI

Para atualizar as propriedades de uma conta de parceiro

O seguinte `update-partner-account` atualiza o `AppServerPrivateKey` para a conta que tem o ID especificado.

```
aws iotwireless update-partner-account \  
  --partner-account-id "78965678771228" \  
  --partner-type "Sidewalk" \  
  --sidewalk  
  AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

Este comando não produz saída.

Para obter mais informações, consulte [Integração do Amazon Sidewalk para AWS IoT Core no Guia](#) de desenvolvedores de AWS IoT.

- Para API obter detalhes, consulte [UpdatePartnerAccount](#) na Referência de AWS CLI Comandos.

update-wireless-device

O código de exemplo a seguir mostra como usar update-wireless-device.

AWS CLI

Para atualizar as propriedades de um dispositivo sem fio

O update-wireless-device exemplo a seguir atualiza as propriedades de um dispositivo sem fio registrado AWS em sua conta.

```
aws iotwireless update-wireless-device \  
  --id "1ffd32c8-8130-4194-96df-622f072a315f" \  
  --destination-name IoTWirelessDestination2 \  
  --description "Using my first LoRaWAN device"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateWirelessDevice](#) na Referência de AWS CLI Comandos.

update-wireless-gateway

O código de exemplo a seguir mostra como usar update-wireless-gateway.

AWS CLI

Para atualizar o gateway sem fio

O update-wireless-gateway exemplo a seguir atualiza a descrição do seu gateway sem fio.

```
aws iotwireless update-wireless-gateway \  
  --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \  
  --description "Using my first LoRaWAN gateway"
```

```
--description "Using my LoRaWAN gateway"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conectando dispositivos e gateways ao AWS IoT Core](#) no Guia do LoRa WAN Desenvolvedor de AWS IoT.

- Para API obter detalhes, consulte [UpdateWirelessGateway](#) na Referência de AWS CLI Comandos.

IVS Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon IVS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-get-channel

O código de exemplo a seguir mostra como usar `batch-get-channel`.

AWS CLI

Para obter informações de configuração de canais sobre vários canais

O `batch-get-channel` exemplo a seguir lista informações sobre os canais especificados.

```
aws ivs batch-get-channel \  
  --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
         arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHiJKl
```


Saída:

```
{
  "channels": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "authorized": false,
      "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
      "insecureIngest": false,
      "latencyMode": "LOW",
      "name": "channel-1",
      "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel-1.abcdEFGH.m3u8",
      "preset": "",
      "playbackRestrictionPolicyArn": "",
      "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh",
      "srt": {
        "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
        "passphrase":
"AB1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
      },
      "tags": {},
      "type": "STANDARD"
    },
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl",
      "authorized": false,
      "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
      "insecureIngest": true,
      "latencyMode": "LOW",
      "name": "channel-2",
      "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel-2.abcdEFGH.m3u8",
      "preset": "",
      "playbackRestrictionPolicyArn": "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/ABCdef34ghIJ",
      "recordingConfigurationArn": "",
      "srt": {
        "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
        "passphrase":
"BA1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
      },
      "tags": {}
    }
  ]
}
```

```

        "type": "STANDARD"
      }
    ]
  }

```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [BatchGetChannel](#) na Referência de AWS CLI Comandos.

batch-get-stream-key

O código de exemplo a seguir mostra como usar batch-get-stream-key.

AWS CLI

Para obter informações sobre várias chaves de transmissão

O batch-get-stream-key exemplo a seguir obtém informações sobre as chaves de stream especificadas.

```

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

```

Saída:

```

{
  "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": {}
    }
  ]
}

```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [BatchGetStreamKey](#) na Referência de AWS CLI Comandos.

batch-start-viewer-session-revocation

O código de exemplo a seguir mostra como usar `batch-start-viewer-session-revocation`.

AWS CLI

Para revogar sessões de visualizador para vários pares de canais ARN e IDs de visualizador

O `batch-start-viewer-session-revocation` exemplo a seguir executa a revogação de sessão em vários pares de ID de canal ARN e visualizador simultaneamente. A solicitação pode ser concluída normalmente, mas retornar valores no campo de erros se o chamador não tiver permissão para revogar a sessão especificada.

```
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}]'
```

Saída:

```
{  
  "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",  
    }  
  ]  
}
```

```
}
```

Para obter mais informações, consulte [Configuração de canais privados](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [BatchStartViewerSessionRevocation](#) na Referência de AWS CLI Comandos.

create-channel

O código de exemplo a seguir mostra como usar `create-channel`.

AWS CLI

Exemplo 1: Para criar um canal sem gravação

O `create-channel` exemplo a seguir cria um novo canal e uma chave de transmissão associada para iniciar a transmissão.

```
aws ivs create-channel \  
  --name "test-channel" \  
  --no-insecure-ingest
```

Saída:

```
{  
  "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": {}
  }
}

```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

Exemplo 2: Para criar um canal com gravação ativada, usando o RecordingConfiguration recurso especificado por seu ARN

O create-channel exemplo a seguir cria um novo canal e uma chave de transmissão associada para iniciar a transmissão e configura a gravação do canal.

```

aws ivs create-channel \
  --name test-channel-with-recording \
  --insecure-ingest \
  --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCD12cdEFgh"

```

Saída:

```

{
  "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": 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": {}
  }
}

```

Para obter mais informações, consulte [Gravar no Amazon S3](#) no Guia do usuário de IVSbaixa latência.

Exemplo 3: Para criar um canal com uma política de restrição de reprodução especificada por seu ARN

O `create-channel` exemplo a seguir cria um novo canal e uma chave de transmissão associada para iniciar a transmissão e configura uma política de restrição de reprodução para o canal.

```

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"

```

Saída:

```

{
  "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":
"AB1C2edfGHijklMN03PqQRstUvwxyzABCDEFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [CreateChannel](#) na Referência de AWS CLI Comandos.

create-playback-restriction-policy

O código de exemplo a seguir mostra como usar `create-playback-restriction-policy`.

AWS CLI

Para criar uma política de restrição de reprodução

O `create-playback-restriction-policy` exemplo a seguir cria uma nova política de restrição de reprodução.

```

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
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [CreatePlaybackRestrictionPolicy](#) na Referência de AWS CLI Comandos.

create-recording-configuration

O código de exemplo a seguir mostra como usar `create-recording-configuration`.

AWS CLI

Para criar um `RecordingConfiguration` recurso

O `create-recording-configuration` exemplo a seguir cria um `RecordingConfiguration` recurso para permitir a gravação no 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}

```

Saída:

```

{
  "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"
      ]
    }
  }
}

```

Para obter mais informações, consulte [Gravar no Amazon S3 no Guia](#) do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateRecordingConfiguration](#) na Referência de AWS CLI Comandos.

create-stream-key

O código de exemplo a seguir mostra como usar `create-stream-key`.

AWS CLI

Para criar uma chave de transmissão

O `create-stream-key` exemplo a seguir cria uma chave de stream para um determinado ARN (Amazon Resource Name).

```
aws ivs create-stream-key \  
--channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Saída:

```
{  
  "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": {}  
  }  
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [CreateStreamKey](#) na Referência de AWS CLI Comandos.

delete-channel

O código de exemplo a seguir mostra como usar `delete-channel`.

AWS CLI

Para excluir um canal e suas chaves de transmissão associadas

O `delete-channel` exemplo a seguir exclui o canal com o especificado ARN (Amazon Resource Name).

```
aws ivs delete-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [DeleteChannel](#) na Referência de AWS CLI Comandos.

delete-playback-key-pair

O código de exemplo a seguir mostra como usar `delete-playback-key-pair`.

AWS CLI

Para excluir um par de chaves de reprodução especificado

O `delete-playback-key-pair` exemplo a seguir retorna a impressão digital do par de chaves especificado.

```
aws ivs delete-playback-key-pair \  
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

Este comando não produz saída.

Para obter mais informações, consulte [Configuração de canais privados](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeletePlaybackKeyPair](#) na Referência de AWS CLI Comandos.

delete-playback-restriction-policy

O código de exemplo a seguir mostra como usar `delete-playback-restriction-policy`.

AWS CLI

Para excluir uma política de restrição de reprodução

O `delete-playback-restriction-policy` exemplo a seguir exclui a política de restrição de reprodução com a política especificada (ARN Amazon Resource Name).

```
aws ivs delete-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABcdef34ghIJ"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [DeletePlaybackRestrictionPolicy](#) na Referência de AWS CLI Comandos.

delete-recording-configuration

O código de exemplo a seguir mostra como usar `delete-recording-configuration`.

AWS CLI

Para excluir o `RecordingConfiguration` recurso especificado por seu ARN

O `delete-recording-configuration` exemplo a seguir exclui o `RecordingConfiguration` recurso com o especificado ARN.

```
aws ivs delete-recording-configuration \  
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gravar no Amazon S3 no Guia](#) do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteRecordingConfiguration](#) na Referência de AWS CLI Comandos.

delete-stream-key

O código de exemplo a seguir mostra como usar `delete-stream-key`.

AWS CLI

Para excluir uma chave de transmissão

O `delete-stream-key` exemplo a seguir exclui a chave de stream de um determinado ARN (Amazon Resource Name), então ela não pode mais ser usada para streaming.

```
aws ivs delete-stream-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [DeleteStreamKey](#) na Referência de AWS CLI Comandos.

get-channel

O código de exemplo a seguir mostra como usar `get-channel`.

AWS CLI

Para obter as informações de configuração de um canal

O `get-channel` exemplo a seguir obtém a configuração do canal para um canal específico ARN (Amazon Resource Name).

```
aws ivs get-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Saída:

```
{  
  "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": {}
  }
}

```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [GetChannel](#) na Referência de AWS CLI Comandos.

get-playback-key-pair

O código de exemplo a seguir mostra como usar `get-playback-key-pair`.

AWS CLI

Para obter um par de chaves de reprodução especificado

O `get-playback-key-pair` exemplo a seguir retorna a impressão digital do par de chaves especificado.

```

aws ivs get-playback-key-pair \
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh

```

Saída:

```

{
  "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": {}
  }
}

```

Para obter mais informações, consulte [Configuração de canais privados](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetPlaybackKeyPair](#) na Referência de AWS CLI Comandos.

get-playback-restriction-policy

O código de exemplo a seguir mostra como usar `get-playback-restriction-policy`.

AWS CLI

Para obter as informações de configuração de uma política de restrição de reprodução

O `get-playback-restriction-policy` exemplo a seguir obtém a configuração da política de restrição de reprodução com a política especificada (ARN Amazon Resource Name).

```
aws ivs get-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABCdef34ghIJ"
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [GetPlaybackRestrictionPolicy](#) na Referência de AWS CLI Comandos.

get-recording-configuration

O código de exemplo a seguir mostra como usar `get-recording-configuration`.

AWS CLI

Para obter informações sobre um `RecordingConfiguration` recurso

O `get-recording-configuration` exemplo a seguir obtém informações sobre o `RecordingConfiguration` recurso especificado ARN.

```
aws ivs get-recording-configuration \  
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCdef34ghIJ"
```

Saída:

```
{  
  "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"
      ]
    }
  }
}

```

Para obter mais informações, consulte [Gravar no Amazon S3 no Guia](#) do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetRecordingConfiguration](#) na Referência de AWS CLI Comandos.

get-stream-key

O código de exemplo a seguir mostra como usar `get-stream-key`.

AWS CLI

Para obter informações sobre um stream

O `get-stream-key` exemplo a seguir obtém informações sobre a chave de stream especificada.

```

aws ivs get-stream-key \
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-
west-2

```

Saída:

```

{
  "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": {}
  }
}

```

```
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [GetStreamKey](#) na Referência de AWS CLI Comandos.

get-stream-session

O código de exemplo a seguir mostra como usar `get-stream-session`.

AWS CLI

Para obter metadados de um fluxo especificado

O `get-stream-session` exemplo a seguir obtém a configuração de metadados do canal especificado ARN (Amazon Resource Name) e do stream especificado; se não `streamId` for fornecido, o stream mais recente do canal será selecionado.

```
aws ivs get-stream-session \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --stream-id "mystream"
```

Saída:

```
{  
  "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":  
        "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
      },  
      "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [GetStreamSession](#) na Referência de AWS CLI Comandos.

get-stream

O código de exemplo a seguir mostra como usar `get-stream`.

AWS CLI

Para obter informações sobre um stream

O `get-stream` exemplo a seguir obtém informações sobre o stream do canal especificado.

```

aws ivs get-stream \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh

```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [GetStream](#) na Referência de AWS CLI Comandos.

import-playback-key-pair

O código de exemplo a seguir mostra como usar `import-playback-key-pair`.

AWS CLI

Para importar a parte pública de um novo par de chaves

O `import-playback-key-pair` exemplo a seguir importa a chave pública especificada (especificada como uma string no PEM formato) e retorna o arn e a impressão digital do novo par de chaves.

```
aws ivs import-playback-key-pair \
  --name "my-playback-key" \
  --public-key-material "G1lbnQxOTA3BgNVBAMMFdoeSBhcmUgew91IGR1..."
```

Saída:

```
{
  "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": {}
  }
}
```

Para obter mais informações, consulte [Configuração de canais privados](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ImportPlaybackKeyPair](#) na Referência de AWS CLI Comandos.

list-channels

O código de exemplo a seguir mostra como usar `list-channels`.

AWS CLI

Exemplo 1: Para obter informações resumidas sobre todos os canais

O `list-channels` exemplo a seguir lista todos os canais AWS da sua conta.

```
aws ivs list-channels
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

Exemplo 2: Para obter informações resumidas sobre todos os canais, filtradas pelo especificado RecordingConfiguration ARN

O `list-channels` exemplo a seguir lista todos os canais AWS da sua conta associados ao especificado RecordingConfiguration ARN.

```

aws ivs list-channels \
  --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"

```

Saída:

```

{
  "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"
    }
  ]
}

```

```
}

```

Para obter mais informações, consulte [Gravar no Amazon S3](#) no Guia do usuário de IVS de baixa latência.

Exemplo 3: Para obter informações resumidas sobre todos os canais, filtradas pelo especificado PlaybackRestrictionPolicy ARN

O `list-channels` exemplo a seguir lista todos os canais AWS da sua conta associados ao especificado PlaybackRestrictionPolicy ARN.

```
aws ivs list-channels \
  --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
  west-2:123456789012:playback-restriction-policy/ABCdef34ghIJ"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [ListChannels](#) na Referência de AWS CLI Comandos.

list-playback-key-pairs

O código de exemplo a seguir mostra como usar `list-playback-key-pairs`.

AWS CLI

Para obter informações resumidas sobre todos os pares de teclas de reprodução

O `list-playback-key-pairs` exemplo a seguir retorna informações sobre todos os pares de chaves.

```
aws ivs list-playback-key-pairs
```

Saída:

```
{
  "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": {}
    }
  ]
}
```

Para obter mais informações, consulte [Configuração de canais privados](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListPlaybackKeyPairs](#) na Referência de AWS CLI Comandos.

`list-playback-restriction-policies`

O código de exemplo a seguir mostra como usar `list-playback-restriction-policies`.

AWS CLI

Para obter informações resumidas sobre todas as políticas de restrição de reprodução

O `list-playback-restriction-policies` exemplo a seguir lista todas as políticas de restrição de reprodução da sua AWS conta.

```
aws ivs list-playback-restriction-policies
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [ListPlaybackRestrictionPolicies](#) na Referência de AWS CLI Comandos.

list-recording-configurations

O código de exemplo a seguir mostra como usar `list-recording-configurations`.

AWS CLI

Para listar todos os RecordingConfiguration recursos criados nessa conta

O `list-recording-configurations` exemplo a seguir obtém informações sobre todos os `RecordingConfiguration` recursos da sua conta.

```
aws ivs list-recording-configurations
```

Saída:

```
{
  "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": {}
    }
  ]
}
```

Para obter mais informações, consulte [Gravar no Amazon S3 no Guia](#) do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListRecordingConfigurations](#) na Referência de AWS CLI Comandos.

list-stream-keys

O código de exemplo a seguir mostra como usar `list-stream-keys`.

AWS CLI

Para obter uma lista de chaves de transmissão

O `list-stream-keys` exemplo a seguir lista todas as chaves de stream de um determinado ARN (Amazon Resource Name).

```
aws ivs list-stream-keys \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Saída:

```
{
  "streamKeys": [
    {
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "tags": {}
    }
  ]
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [ListStreamKeys](#) na Referência de AWS CLI Comandos.

list-stream-sessions

O código de exemplo a seguir mostra como usar `list-stream-sessions`.

AWS CLI

Para obter um resumo dos fluxos atuais e anteriores de um canal específico na região atual AWS

O `list-stream-sessions` exemplo a seguir relata informações resumidas de streams de um canal específico 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 ""
```

Saída:

```
{  
  "nextToken": "set-2",  
  "streamSessions": [  
    {  
      "startTime": 1641578182,  
      "endTime": 1641579982,  
      "hasErrorEvent": false,  
      "streamId": "mystream"  
    }  
    ...  
  ]  
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [ListStreamSessions](#) na Referência de AWS CLI Comandos.

list-streams

O código de exemplo a seguir mostra como usar `list-streams`.

AWS CLI

Para obter uma lista de transmissões ao vivo e seu estado

O `list-streams` exemplo a seguir lista todas as transmissões ao vivo da sua AWS conta.

```
aws ivs list-streams
```

Saída:

```
{  
  "streams": [  
    {
```

```
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "state": "LIVE",
        "health": "HEALTHY",
        "streamId": "st-ABCDEFghij01234KLMN5678",
        "viewerCount": 1
    }
]
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [ListStreams](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar todas as tags de um AWS recurso (por exemplo: canal, chave de transmissão)

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um recurso específico ARN (Amazon Resource Name).

```
aws ivs list-tags-for-resource \
  --resource-arn arn:aws:ivs:us-west-2:12345689012:channel/abcdABCDefgh
```

Saída:

```
{
  "tags":
  {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Para obter mais informações, consulte [Marcação](#) na API Referência do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-metadata

O código de exemplo a seguir mostra como usar `put-metadata`.

AWS CLI

Para inserir metadados no fluxo ativo de um canal especificado

O `put-metadata` exemplo a seguir insere os metadados fornecidos no fluxo do canal especificado.

```
aws ivs put-metadata \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --metadata '{"my": "metadata"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [PutMetadata](#) na Referência de AWS CLI Comandos.

start-viewer-session-revocation

O código de exemplo a seguir mostra como usar `start-viewer-session-revocation`.

AWS CLI

Para revogar uma sessão de espectador para um determinado par de vários canais ARN e IDs de visualizador

O `start-viewer-session-revocation` exemplo a seguir inicia o processo de revogação da sessão do espectador associada a um canal ARN e ID de visualizador especificados, incluindo o número da versão da sessão especificada. Se a versão não for fornecida, o padrão é 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Configuração de canais privados](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [StartViewerSessionRevocation](#) na Referência de AWS CLI Comandos.

stop-stream

O código de exemplo a seguir mostra como usar `stop-stream`.

AWS CLI

Para interromper um fluxo especificado

O `stop-stream` exemplo a seguir interrompe a transmissão no canal especificado.

```
aws ivs stop-stream \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [StopStream](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar ou atualizar tags para um AWS recurso (por exemplo: canal, chave de transmissão)

O `tag-resource` exemplo a seguir adiciona ou atualiza tags para um recurso específico ARN (Amazon Resource Name).

```
aws ivs tag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) na API Referência do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um AWS recurso (por exemplo: canal, chave de transmissão)

O `untag-resource` exemplo a seguir remove as tags especificadas para um recurso específico ARN (Amazon Resource Name).

```
aws ivs untag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tag-keys "tagkey1, tagkey2"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) na API Referência do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-channel

O código de exemplo a seguir mostra como usar `update-channel`.

AWS CLI

Exemplo 1: Para atualizar as informações de configuração de um canal

O `update-channel` exemplo a seguir atualiza a configuração do canal de um canal especificado ARN para alterar o nome do canal. Isso não afeta um fluxo contínuo desse canal; você deve parar e reiniciar o fluxo para que as alterações entrem em vigor.

```
aws ivs update-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --name "channel-1" \  
  --insecure-ingest
```

Saída:

```
{
  "channel": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
    "name": "channel-1",
    "latencyMode": "LOW",
    "type": "STANDARD",
    "playbackRestrictionPolicyArn": "",
    "recordingConfigurationArn": "",
    "srt": {
      "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
      "passphrase":
"AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
    "insecureIngest": true,
    "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
    "preset": "",
    "authorized": false,
    "tags": {}
  }
}
```

Para obter mais informações, consulte [Criar um canal](#) no Guia do usuário IVS de baixa latência.

Exemplo 2: Para atualizar a configuração de um canal para ativar a gravação

O `update-channel` exemplo a seguir atualiza a configuração do canal de um canal específico ARN para permitir a gravação. Isso não afeta um fluxo contínuo desse canal; você deve parar e reiniciar o fluxo para que as alterações entrem em vigor.

```
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"
```

Saída:

```
{
  "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": {}
  }
}

```

Para obter mais informações, consulte [Gravar no Amazon S3](#) no Guia do usuário de IVSbaixa latência.

Exemplo 3: Para atualizar a configuração de um canal para desativar a gravação

O `update-channel` exemplo a seguir atualiza a configuração do canal de um canal específico ARN para desativar a gravação. Isso não afeta um fluxo contínuo desse canal; você deve parar e reiniciar o fluxo para que as alterações entrem em vigor.

```

aws ivs update-channel \
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
  --recording-configuration-arn ""

```

Saída:

```

{
  "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":
"AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijk1MN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Para obter mais informações, consulte [Gravar no Amazon S3](#) no Guia do usuário de IVS baixa latência.

Exemplo 4: Para atualizar a configuração de um canal para ativar a restrição de reprodução

O `update-channel` exemplo a seguir atualiza a configuração do canal de um canal específico ARN para aplicar uma política de restrição de reprodução. Isso não afeta um fluxo contínuo desse canal; você deve parar e reiniciar o fluxo para que as alterações entrem em vigor.

```

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"

```

Saída:

```

{
  "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":
"AB1C2defGHijklMN03PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

Exemplo 5: Para atualizar a configuração de um canal para desativar a restrição de reprodução

O `update-channel` exemplo a seguir atualiza a configuração do canal de um canal específico ARN para desativar a restrição de reprodução. Isso não afeta um fluxo contínuo desse canal; você deve parar e reiniciar o fluxo para que as alterações entrem em vigor.

```

aws ivs update-channel \
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
  --playback-restriction-policy-arn ""

```

Saída:

```

{
  "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":
"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",
    "preset": "",
    "authorized": false,
    "tags": {}
  }
}

```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [UpdateChannel](#) na Referência de AWS CLI Comandos.

update-playback-restriction-policy

O código de exemplo a seguir mostra como usar `update-playback-restriction-policy`.

AWS CLI

Para atualizar uma política de restrição de reprodução

O `update-playback-restriction-policy` exemplo a seguir atualiza a política de restrição de reprodução com a política especificada ARN para desativar a imposição estrita de origem. Isso não afeta um fluxo contínuo do canal associado; você deve parar e reiniciar o fluxo para que as alterações entrem em vigor.

```

aws ivs update-playback-restriction-policy \
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ" \
  --no-enable-strict-origin-enforcement

```

Saída:

```

{
  "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"
  }
}
```

Para obter mais informações, consulte [Conteúdo e visualizadores indesejados no Guia](#) do usuário IVS de baixa latência.

- Para API obter detalhes, consulte [UpdatePlaybackRestrictionPolicy](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon IVS Chat usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon IVS Chat.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-chat-token

O código de exemplo a seguir mostra como usar `create-chat-token`.

AWS CLI

Para criar um token de bate-papo

O `create-chat-token` exemplo a seguir cria um token de bate-papo criptografado que é usado para estabelecer uma WebSocket conexão individual com uma sala. O token é válido por um minuto e uma conexão (sessão) estabelecida com o token é válida pela duração especificada.

```
aws ivschat create-chat-token \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \  
  --userId "11231234" \  
  --capabilities "SEND_MESSAGE", \  
  --sessionDurationInMinutes 30
```

Saída:

```
{  
  "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",  
  "sessionExpirationTime": "2022-03-16T04:44:09+00:00"  
  "state": "CREATING",  
  "tokenExpirationTime": "2022-03-16T03:45:09+00:00"  
}
```

Para obter mais informações, consulte [Etapa 3: Autenticar e autorizar clientes de chat no Guia](#) do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateChatToken](#) na Referência de AWS CLI Comandos.

create-logging-configuration

O código de exemplo a seguir mostra como usar `create-logging-configuration`.

AWS CLI

Para criar um LoggingConfiguration recurso de bate-papo

O `create-logging-configuration` exemplo a seguir cria um LoggingConfiguration recurso que permite aos clientes armazenar e registrar as mensagens enviadas.

```
aws ivschat create-logging-configuration \  
  --destination-configuration s3={bucketName=demo-logging-bucket} \  
  --name "test-logging-config" \  
  --
```



```
--tags "key1=value1, key2=value2"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateLoggingConfiguration](#) na Referência de AWS CLI Comandos.

create-room

O código de exemplo a seguir mostra como usar `create-room`.

AWS CLI

Para criar uma sala

O `create-room` exemplo a seguir cria uma nova sala.

```
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
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Etapa 2: Criar uma sala de bate-papo](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateRoom](#) na Referência de AWS CLI Comandos.

delete-logging-configuration

O código de exemplo a seguir mostra como usar delete-logging-configuration.

AWS CLI

Para excluir um LoggingConfiguration recurso de bate-papo

O delete-logging-configuration exemplo a seguir exclui o LoggingConfiguration recurso para o especificadoARN.

```
aws ivschat delete-logging-configuration \
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
  ABcdef34ghIJ"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteLoggingConfiguration](#) na Referência de AWS CLI Comandos.

delete-message

O código de exemplo a seguir mostra como usar delete-message.

AWS CLI

Para excluir mensagens de uma sala específica

O delete-message exemplo a seguir envia um evento para a sala especificada, o que direciona os clientes a excluírem a mensagem especificada: ou seja, retirá-la da exibição e excluí-la do histórico de bate-papo do cliente.

```
aws ivschat delete-message \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --id "ABC123def456" \  
  --reason "Message contains profanity"
```

Saída:

```
{  
  "id": "12345689012"  
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteMessage](#) na Referência de AWS CLI Comandos.

delete-room

O código de exemplo a seguir mostra como usar delete-room.

AWS CLI

Para excluir uma sala

O delete-room exemplo a seguir exclui a sala especificada. Os clientes conectados são desconectados. Em caso de sucesso, ele retorna HTTP 204 com um corpo de resposta vazio.

```
aws ivschat delete-room \  
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteRoom](#) na Referência de AWS CLI Comandos.

disconnect-user

O código de exemplo a seguir mostra como usar `disconnect-user`.

AWS CLI

Para desconectar um usuário de uma sala

O `disconnect-user` exemplo a seguir desconecta todas as conexões do usuário especificado da sala especificada. Em caso de sucesso, ele retorna HTTP 200 com um corpo de resposta vazio.

```
aws ivschat disconnect-user \  
  --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --userId "ABC123def456" \  
  --reason "Violated terms of service"
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DisconnectUser](#) na Referência de AWS CLI Comandos.

get-logging-configuration

O código de exemplo a seguir mostra como usar `get-logging-configuration`.

AWS CLI

Para obter informações sobre um LoggingConfiguration recurso

O `get-logging-configuration` exemplo a seguir obtém informações sobre o LoggingConfiguration recurso especificadoARN.

```
aws ivschat get-logging-configuration \  
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABcdef34ghIJ"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetLoggingConfiguration](#) na Referência de AWS CLI Comandos.

get-room

O código de exemplo a seguir mostra como usar `get-room`.

AWS CLI

Para obter o quarto especificado

O `get-room` exemplo a seguir obtém informações sobre a sala especificada.

```
aws ivschat get-room \  
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetRoom](#) na Referência de AWS CLI Comandos.

list-logging-configurations

O código de exemplo a seguir mostra como usar `list-logging-configurations`.

AWS CLI

Para obter informações resumidas sobre todas as configurações de registro para o usuário na AWS região em que a API solicitação é processada

O `list-logging-configurations` exemplo a seguir lista informações sobre todos os `LoggingConfiguration` recursos do usuário na AWS região em que a API solicitação é processada.

```
aws ivschat list-logging-configurations \
  --max-results 2 \
  --next-token ""
```

Saída:

```
{
  "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"
}
...
]
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListLoggingConfigurations](#) na Referência de AWS CLI Comandos.

list-rooms

O código de exemplo a seguir mostra como usar `list-rooms`.

AWS CLI

Para obter informações resumidas sobre todos os seus quartos na região atual

O `list-rooms` exemplo a seguir obtém informações resumidas sobre todas as salas na AWS região em que a solicitação é processada. Os resultados são classificados em ordem decrescente de `updateTime`

```
aws ivschat list-rooms \
  --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
  --max-results 10 \
  --next-token ""
```

Saída:

```
{
  "nextToken": "page3",
  "rooms": [
    {
      "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",
      "createTime": "2022-03-16T04:44:09+00:00",
      "id": "g1H2I3j4k5L6",
      "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-west-2:123456789012:logging-configuration/ABCdef34ghIJ"],
      "name": "test-room-1",
      "tags": {},
      "updateTime": "2022-03-16T07:22:09+00:00"
    }
  ]
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListRooms](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar todas as tags de um AWS recurso (por exemplo: Room)

O `list-tags-for-resource` exemplo a seguir lista todas as tags de um recurso específico ARN (Amazon Resource Name).

```
aws ivschat list-tags-for-resource \
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6
```

Saída:

```
{
  "tags":
  {
```



```
    "key1": "value1",  
    "key2": "value2"  
  }  
}
```

Para obter mais informações, consulte [Marcação](#) na APIReferência do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

send-event

O código de exemplo a seguir mostra como usar send-event.

AWS CLI

Para enviar um evento para uma sala

O send-event exemplo a seguir envia o evento fornecido para a sala especificada.

```
aws ivschat send-event \  
  --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --eventName "SystemMessage" \  
  --attributes \  
    "msgType"="user-notification", \  
    "msgText"="This chat room will close in 15 minutes."
```

Saída:

```
{  
  "id": "12345689012"  
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [SendEvent](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para adicionar ou atualizar tags para um AWS recurso (por exemplo: Sala)

O `tag-resource` exemplo a seguir adiciona ou atualiza tags para um recurso específico ARN (Amazon Resource Name). Em caso de sucesso, ele retorna HTTP 200 com um corpo de resposta vazio.

```
aws ivschat tag-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
  --tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) na APIReferência do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um AWS recurso (por exemplo: Sala)

O `untag-resource` exemplo a seguir remove as tags especificadas para um recurso específico ARN (Amazon Resource Name). Em caso de sucesso, ele retorna HTTP 200 com um corpo de resposta vazio.

```
aws ivschat untag-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
  --tag-keys "tagkey1, tagkey2"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcação](#) na APIReferência do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-logging-configuration

O código de exemplo a seguir mostra como usar `update-logging-configuration`.

AWS CLI

Para atualizar a configuração de registro de uma sala

O `update-logging-configuration` exemplo a seguir atualiza um `LoggingConfiguration` recurso com os dados fornecidos.

```
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"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [UpdateLoggingConfiguration](#) na Referência de AWS CLI Comandos.

update-room

O código de exemplo a seguir mostra como usar `update-room`.

AWS CLI

Para atualizar a configuração de uma sala

O `update-room` exemplo a seguir atualiza a configuração da sala especificada com os dados fornecidos.

```
aws ivschat update-room \  
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --logging-configuration-identifiers "arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \  
  --name "chat-room-a" \  
  --maximum-message-length 256 \  
  --maximum-message-rate-per-second 5
```

Saída:

```
{  
  "arn": "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6",  
  "createTime": "2022-03-16T04:44:09+00:00",  
  "id": "g1H2I3j4k5L6",  
  "loggingConfigurationIdentifiers": ["arn:aws:ivschat:us-  
west-2:123456789012:logging-configuration/ABcdef34ghIJ"],  
  "maximumMessageLength": 256,  
  "maximumMessageRatePerSecond": 5,  
  "name": "chat-room-a",  
  "tags": {},  
  "updateTime": "2022-03-16T07:22:09+00:00"  
}
```

Para obter mais informações, consulte [Conceitos básicos do Amazon IVS Chat](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [UpdateRoom](#) na Referência de AWS CLI Comandos.

Exemplos de streaming IVS em tempo real da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface Amazon IVS Real-Time Streaming.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-encoder-configuration

O código de exemplo a seguir mostra como usar `create-encoder-configuration`.

AWS CLI

Para criar uma configuração de codificador de composição

O `create-encoder-configuration` exemplo a seguir cria uma configuração de codificador de composição com as propriedades especificadas.

```
aws ivs-realtime create-encoder-configuration \  
  --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

Saída:

```
{  
  "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  
    }  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateEncoderConfiguration](#) na Referência de AWS CLI Comandos.

create-participant-token

O código de exemplo a seguir mostra como usar `create-participant-token`.

AWS CLI

Para criar um token de participante do estágio

O `create-participant-token` exemplo a seguir cria um token de participante para o estágio especificado.

```
aws ivs-realtime create-participant-token \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --user-id bob
```

Saída:

```
{  
  "participantToken": {  
    "expirationTime": "2023-03-07T09:47:43+00:00",  
    "participantId": "ABCDEFghij01234KLMN6789",  
    "token": "abcd1234defg5678"  
  }  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateParticipantToken](#) na Referência de AWS CLI Comandos.

create-stage

O código de exemplo a seguir mostra como usar `create-stage`.

AWS CLI

Para criar um palco

O `create-stage` exemplo a seguir cria um token de estágio e participante do estágio para um usuário especificado.

```
aws ivs-realtime create-stage \  
  --name stage1 \  
  --participant-token-configurations userId=alice
```

Saída:

```
{  
  "participantTokens": [  
    {  
      "participantId": "ABCDEFghij01234KLMN5678",  
      "token": "a1b2c3d4567890ab",  
      "userId": "alice"  
    }  
  ],  
  "stage": {  
    "activeSessionId": "st-a1b2c3d4e5f6g",  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",  
    "name": "stage1",  
    "tags": {}  
  }  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateStage](#) na Referência de AWS CLI Comandos.

create-storage-configuration

O código de exemplo a seguir mostra como usar `create-storage-configuration`.

AWS CLI

Para criar uma configuração de armazenamento de composição

O `create-storage-configuration` exemplo a seguir cria uma configuração de armazenamento de composição com as propriedades especificadas.

```
aws ivs-realtime create-storage-configuration \  
  --name "test-sc" --s3 "bucketName=test-bucket-name"
```

Saída:

```
{  
  "storageConfiguration": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
ABabCDcdEFef",  
    "name": "test-sc",  
    "s3": {  
      "bucketName": "test-bucket-name"  
    },  
    "tags": {}  
  }  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [CreateStorageConfiguration](#) na Referência de AWS CLI Comandos.

delete-encoder-configuration

O código de exemplo a seguir mostra como usar `delete-encoder-configuration`.

AWS CLI

Para excluir uma configuração de codificador de composição

O seguinte `delete-encoder-configuration` exclui a configuração do codificador de composição especificada pelo dado (ARN Amazon Resource Name).

```
aws ivs-realtime delete-encoder-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/  
  ABabCDcdEFef"
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteEncoderConfiguration](#) na Referência de AWS CLI Comandos.

delete-stage

O código de exemplo a seguir mostra como usar `delete-stage`.

AWS CLI

Para excluir um estágio

O `delete-stage` exemplo a seguir exclui o estágio especificado.

```
aws ivs-realtime delete-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteStage](#) na Referência de AWS CLI Comandos.

delete-storage-configuration

O código de exemplo a seguir mostra como usar `delete-storage-configuration`.

AWS CLI

Para excluir uma configuração de armazenamento de composição

O seguinte `delete-storage-configuration` exclui a configuração de armazenamento da composição especificada pelo dado ARN (Amazon Resource Name).

```
aws ivs-realtime delete-storage-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
  ABabCDcdEFef"
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DeleteStorageConfiguration](#) na Referência de AWS CLI Comandos.

disconnect-participant

O código de exemplo a seguir mostra como usar `disconnect-participant`.

AWS CLI

Para desconectar um participante da etapa

O `disconnect-participant` exemplo a seguir desconecta o participante especificado do estágio especificado.

```
aws ivs-realtime disconnect-participant \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --participant-id ABCDEfghij01234KLMN5678
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [DisconnectParticipant](#) na Referência de AWS CLI Comandos.

get-composition

O código de exemplo a seguir mostra como usar `get-composition`.

AWS CLI

Exemplo 1: Para obter uma composição com configurações de layout padrão

O `get-composition` exemplo a seguir obtém a composição do ARN (Amazon Resource Name) especificado.

```
aws ivs-realtime get-composition \  
--arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Saída:

```
{  
  "composition": {  
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",  
    "destinations": [  
      {  
        "configuration": {  
          "channel": {  
            "channelArn": "arn:aws:ivs:ap-northeast-1:123456789012:channel/abcABCdefDEg",  
            "encoderConfigurationArn": "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"  
          },  
          "name": ""  
        },  
        "id": "AabBCcdDEefF",  
        "startTime": "2023-10-16T23:26:00+00:00",  
        "state": "ACTIVE"  
      },  
      {  
        "configuration": {  
          "name": "",  
          "s3": {  
            "encoderConfigurationArns": [  
              "arn:aws:ivs:arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"  
            ],  
            "recordingConfiguration": {  
              "format": "HLS"  
            },  
            "storageConfigurationArn": "arn:arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/FefABabCDcdE"  
          }  
        }  
      }  
    ]  
  }  
}
```

```

        }
      },
      "detail": {
        "s3": {
          "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/"
        }
      },
      "id": "GHFabcgefABC",
      "startTime": "2023-10-16T23:26:00+00:00",
      "state": "STARTING"
    }
  ],
  "layout": {
    "grid": {
      "featuredParticipantAttribute": ""
      "gridGap": 2,
      "omitStoppedVideo": false,
      "videoAspectRatio": "VIDEO",
      "videoFillMode": ""
    }
  },
  "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
  "startTime": "2023-10-16T23:24:00+00:00",
  "state": "ACTIVE",
  "tags": {}
}
}

```

Para obter mais informações, consulte [Gravação composta \(streaming em tempo real\)](#) no Guia do usuário do Amazon Interactive Video Service.

Exemplo 2: Para obter uma composição com layout PiP

O `get-composition` exemplo a seguir obtém a composição do ARN (Amazon Resource Name) especificado, que usa o layout PiP.

```

aws ivs-realtime get-composition \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"

```

Saída:

```

{
  "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": {}
}
```

Para obter mais informações, consulte [Gravação composta \(streaming em tempo real\)](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetComposition](#) na Referência de AWS CLI Comandos.

get-encoder-configuration

O código de exemplo a seguir mostra como usar `get-encoder-configuration`.

AWS CLI

Para obter uma configuração de codificador de composição

O `get-encoder-configuration` exemplo a seguir obtém a configuração do codificador de composição especificada pelo dado ARN (Amazon Resource Name).

```
aws ivs-realtime get-encoder-configuration \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

Saída:

```
{
  "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
    }
  }
}

```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetEncoderConfiguration](#) na Referência de AWS CLI Comandos.

get-participant

O código de exemplo a seguir mostra como usar `get-participant`.

AWS CLI

Para conseguir um participante da etapa

O `get-participant` exemplo a seguir obtém o participante do estágio para um ID de participante e ID de sessão especificados no estágio especificado 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

```

Saída:

```

{
  "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,
"sdkVersion", "",
"state": "DISCONNECTED",
"userId": ""
}
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetParticipant](#) na Referência de AWS CLI Comandos.

get-stage-session

O código de exemplo a seguir mostra como usar `get-stage-session`.

AWS CLI

Para obter uma sessão de palco

O `get-stage-session` exemplo a seguir obtém a sessão de estágio para um ID de sessão especificado de um estágio específico 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
```

Saída:

```
{
  "stageSession": {
    "endTime": "2023-04-26T20:36:29+00:00",
    "sessionId": "st-a1b2c3d4e5f6g",
    "startTime": "2023-04-26T20:30:29.602000+00:00"
  }
}
```



```
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetStageSession](#) na Referência de AWS CLI Comandos.

get-stage

O código de exemplo a seguir mostra como usar `get-stage`.

AWS CLI

Para obter as informações de configuração de um estágio

O `get-stage` exemplo a seguir obtém a configuração do estágio para um estágio específico ARN (Amazon Resource Name).

```
aws ivs-realtime get-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Saída:

```
{  
  "stage": {  
    "activeSessionId": "st-a1b2c3d4e5f6g",  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",  
    "name": "test",  
    "tags": {}  
  }  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetStage](#) na Referência de AWS CLI Comandos.

get-storage-configuration

O código de exemplo a seguir mostra como usar `get-storage-configuration`.

AWS CLI

Para obter uma configuração de armazenamento de composição

O `get-storage-configuration` exemplo a seguir obtém a configuração de armazenamento da composição especificada pelo dado ARN (Amazon Resource Name).

```
aws ivs-realtime get-storage-configuration \
  --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
  abcdABCDefgh"
```

Saída:

```
{
  "storageConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
    "name": "test-sc",
    "s3": {
      "bucketName": "test-bucket-name"
    },
    "tags": {}
  }
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [GetStorageConfiguration](#) na Referência de AWS CLI Comandos.

list-compositions

O código de exemplo a seguir mostra como usar `list-compositions`.

AWS CLI

Para obter uma lista de composições

A seguir, são `list-compositions` listadas todas as composições AWS da sua conta, na AWS região em que a API solicitação é processada.

aws ivs-realtime list-compositions

Saída:

```
{
  "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": {}
    }
]
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListCompositions](#) na Referência de AWS CLI Comandos.

list-encoder-configurations

O código de exemplo a seguir mostra como usar `list-encoder-configurations`.

AWS CLI

Para listar as configurações do codificador de composição

Veja a seguir `list-encoder-configurations` todas as configurações do codificador de composição da sua AWS conta, na AWS região em que a API solicitação é processada.

```
aws ivs-realtime list-encoder-configurations
```

Saída:

```
{
  "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": {}
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListEncoderConfigurations](#) na Referência de AWS CLI Comandos.

list-participant-events

O código de exemplo a seguir mostra como usar `list-participant-events`.

AWS CLI

Para obter uma lista dos eventos dos participantes do palco

O `list-participant-events` exemplo a seguir lista todos os eventos de participantes para um ID de participante e ID de sessão específicos de um estágio específico 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
```

Saída:

```
{
  "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"
  }
]
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListParticipantEvents](#) na Referência de AWS CLI Comandos.

list-participants

O código de exemplo a seguir mostra como usar `list-participants`.

AWS CLI

Para obter uma lista dos participantes da etapa

O `list-participants` exemplo a seguir lista todos os participantes de um ID de sessão específico de um estágio específico ARN (Amazon Resource Name).

```
aws ivs-realtime list-participants \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g
```

Saída:

```
{
  "participants": [
    {
      "firstJoinTime": "2023-04-26T20:30:34+00:00",
      "participantId": "abCDEf12GHIj"
      "published": true,
      "state": "DISCONNECTED",
      "userId": ""
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListParticipants](#) na Referência de AWS CLI Comandos.

list-stage-sessions

O código de exemplo a seguir mostra como usar `list-stage-sessions`.

AWS CLI

Para obter uma lista das sessões de palco

O `list-stage-sessions` exemplo a seguir lista todas as sessões de um estágio específico ARN (Amazon Resource Name).

```
aws ivs-realtime list-stage-sessions \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Saída:

```
{  
  "stageSessions": [  
    {  
      "endTime": "2023-04-26T20:36:29+00:00",  
      "sessionId": "st-a1b2c3d4e5f6g",  
      "startTime": "2023-04-26T20:30:29.602000+00:00"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListStageSessions](#) na Referência de AWS CLI Comandos.

list-stages

O código de exemplo a seguir mostra como usar `list-stages`.

AWS CLI

Para obter informações resumidas sobre todas as etapas

O `list-stages` exemplo a seguir lista todos os estágios AWS da sua conta, na AWS região em que a API solicitação é processada.

```
aws ivs-realtime list-stages
```

Saída:

```
{
  "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": {}
    }
  ]
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListStages](#) na Referência de AWS CLI Comandos.

list-storage-configurations

O código de exemplo a seguir mostra como usar `list-storage-configurations`.

AWS CLI

Para listar as configurações de armazenamento de composição

A seguir, são `list-storage-configurations` listadas todas as configurações de armazenamento de composição AWS da sua conta, na AWS região em que a API solicitação é processada.

```
aws ivs-realtime list-storage-configurations
```

Saída:

```
{
  "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": {}
    }
  ]
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [ListStorageConfigurations](#) na Referência de AWS CLI Comandos.

start-composition

O código de exemplo a seguir mostra como usar `start-composition`.

AWS CLI

Exemplo 1: Para iniciar uma composição com configurações de layout padrão

O `start-composition` exemplo a seguir inicia uma composição para o estágio especificado a ser transmitido para os locais especificados.

```
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"}}]'
```

Saída:

```
{
  "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": {}
}
}

```

Para obter mais informações, consulte [Gravação composta \(streaming em tempo real\)](#) no Guia do usuário do Amazon Interactive Video Service.

Exemplo 2: Para iniciar uma composição com o layout PiP

O `start-composition` exemplo a seguir inicia uma composição para o estágio especificado a ser transmitido para os locais especificados usando o 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}"'
```

Saída:

```
{
  "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": {}
}
}

```

Para obter mais informações, consulte [Gravação composta \(streaming em tempo real\)](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [StartComposition](#) na Referência de AWS CLI Comandos.

stop-composition

O código de exemplo a seguir mostra como usar stop-composition.

AWS CLI

Para parar uma composição

O seguinte `stop-composition` interrompe a composição especificada pelo determinado ARN (Amazon Resource Name).

```
aws ivs-realtime stop-composition \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [StopComposition](#) na Referência de AWS CLI Comandos.

update-stage

O código de exemplo a seguir mostra como usar `update-stage`.

AWS CLI

Para atualizar a configuração de um estágio

O `update-stage` exemplo a seguir atualiza um estágio para um estágio especificado ARN para atualizar o nome do estágio.

```
aws ivs-realtime update-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --name stage1a
```

Saída:

```
{  
  "stage": {  
    "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",  
    "name": "stage1a"  
  }  
}
```

Para obter mais informações, consulte [Habilitando vários hosts em um Amazon IVS Stream](#) no Guia do usuário do Amazon Interactive Video Service.

- Para API obter detalhes, consulte [UpdateStage](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Kendra usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon Kendra.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-data-source

O código de exemplo a seguir mostra como usar create-data-source.

AWS CLI

Para criar um conector de fonte de dados Amazon Kendra

O seguinte create-data-source cria e configura um conector de fonte de dados Amazon Kendra. Você pode usar describe-data-source para visualizar o status de um conector de fonte de dados e ler qualquer mensagem de erro se o status mostrar um conector de fonte de dados FAILED "" para criar 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"]}

```

Saída:

```

{
  "Id": "exampledatasource1"
}

```

Para obter mais informações, consulte [Introdução a um índice e conector de fonte de dados do Amazon Kendra](#) no Guia do desenvolvedor do Amazon Kendra.

- Para API obter detalhes, consulte [CreateDataSource](#) na Referência de AWS CLI Comandos.

create-index

O código de exemplo a seguir mostra como usar create-index.

AWS CLI

Para criar um índice do Amazon Kendra

O seguinte create-index cria e configura um índice do Amazon Kendra. Você pode usar describe-index para visualizar o status de um índice e ler qualquer mensagem de erro se o status mostrar um índice "FAILED" para criar 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"}}'
```

Saída:

```
{
  "Id": index1
}
```

Para obter mais informações, consulte [Introdução a um índice e conector de fonte de dados do Amazon Kendra](#) no Guia do desenvolvedor do Amazon Kendra.

- Para API obter detalhes, consulte [CreateIndex](#) na Referência de AWS CLI Comandos.

describe-data-source

O código de exemplo a seguir mostra como usar `describe-data-source`.

AWS CLI

Para obter informações sobre um conector de fonte de dados Amazon Kendra

Veja a seguir `describe-data-source` informações sobre um conector de fonte de dados Amazon Kendra. Você pode visualizar a configuração de um conector de fonte de dados e ler qualquer mensagem de erro se o status mostrar um conector de fonte de dados FAILED "" para criar completamente.

```
aws kendra describe-data-source \
  --id exampledatasource1 \
  --index-id exampleindex1
```

Saída:

```
{
  "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"]
    }
  }
}

```

Para obter mais informações, consulte [Introdução a um índice e conector de fonte de dados do Amazon Kendra](#) no Guia do desenvolvedor do Amazon Kendra.

- Para API obter detalhes, consulte [DescribeDataSource](#) na Referência de AWS CLI Comandos.

describe-index

O código de exemplo a seguir mostra como usar `describe-index`.

AWS CLI

Para obter informações sobre um índice da Amazon Kendra

Veja a seguir informações sobre um índice da Amazon `describe-index` Kendra. Você pode visualizar a configuração de um índice e ler qualquer mensagem de erro se o status mostrar um índice "FAILED" para criar completamente.


```
        "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"
    }
  }
]
}

```

Para obter mais informações, consulte [Introdução a um índice e conector de fonte de dados do Amazon Kendra](#) no Guia do desenvolvedor do Amazon Kendra.

- Para API obter detalhes, consulte [DescribeIndex](#) na Referência de AWS CLI Comandos.

update-data-source

O código de exemplo a seguir mostra como usar `update-data-source`.

AWS CLI

Para atualizar um conector de fonte de dados do Amazon Kendra

O seguinte `update-data-source` atualiza a configuração de um conector de fonte de dados Amazon Kendra. Se a ação for bem-sucedida, o serviço não retornará nenhuma saída, o código de HTTP status 200 ou o código de AWS CLI retorno 0. Você pode usar `describe-data-source` para visualizar a configuração e o status de um conector de fonte de dados.

```

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"]}

```

Este comando não produz saída.

Para obter mais informações, consulte [Introdução a um índice e conector de fonte de dados do Amazon Kendra](#) no Guia do desenvolvedor do Amazon Kendra.

- Para API obter detalhes, consulte [UpdateDataSource](#) na Referência de AWS CLI Comandos.

update-index

O código de exemplo a seguir mostra como usar update-index.

AWS CLI

Para atualizar um índice do Amazon Kendra

O seguinte update-index atualiza a configuração de um índice do Amazon Kendra. Se a ação for bem-sucedida, o serviço não retornará nenhuma saída, o código de HTTP status 200 ou o código de AWS CLI retorno 0. Você pode usar describe-index para visualizar a configuração e o status de um índice.

```

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"}'

```

Este comando não produz saída.

Para obter mais informações, consulte [Introdução a um índice e conector de fonte de dados do Amazon Kendra](#) no Guia do desenvolvedor do Amazon Kendra.

- Para API obter detalhes, consulte [UpdateIndex](#) na Referência de AWS CLI Comandos.

Exemplos de Kinesis usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Kinesis.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-stream

O código de exemplo a seguir mostra como usar `add-tags-to-stream`.

AWS CLI

Para adicionar tags a um fluxo de dados

O `add-tags-to-stream` exemplo a seguir atribui uma tag com a chave `samplekey` e o valor `example` ao fluxo especificado.

```
aws kinesis add-tags-to-stream \  
  --stream-name samplestream \  
  --tags samplekey=example
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus streams no Guia](#) do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [AddTagsToStream](#) na Referência de AWS CLI Comandos.

create-stream

O código de exemplo a seguir mostra como usar `create-stream`.

AWS CLI

Como criar um fluxo de dados

O exemplo de `create-stream` a seguir cria um fluxo de dados chamado `samplestream` com três fragmentos.

```
aws kinesis create-stream \  
  --stream-name samplestream \  
  --shard-count 3
```

Este comando não produz saída.

Para obter mais informações, consulte [Creating a stream](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [CreateStream](#) na Referência de AWS CLI Comandos.

decrease-stream-retention-period

O código de exemplo a seguir mostra como usar `decrease-stream-retention-period`.

AWS CLI

Para diminuir o período de retenção do fluxo de dados

O `decrease-stream-retention-period` exemplo a seguir diminui o período de retenção (o tempo em que os registros de dados ficam acessíveis após serem adicionados ao stream) de um stream chamado `samplestream` para 48 horas.

```
aws kinesis decrease-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 48
```

Este comando não produz saída.

Para obter mais informações, consulte [Alteração do período de retenção de dados](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DecreaseStreamRetentionPeriod](#) na Referência de AWS CLI Comandos.

delete-stream

O código de exemplo a seguir mostra como usar `delete-stream`.

AWS CLI

Como excluir um fluxo de dados

O exemplo de `delete-stream` a seguir exclui o fluxo de dados especificado.

```
aws kinesis delete-stream \  
  --stream-name samplestream
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir um stream](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DeleteStream](#) na Referência de AWS CLI Comandos.

deregister-stream-consumer

O código de exemplo a seguir mostra como usar `deregister-stream-consumer`.

AWS CLI

Para cancelar o registro de um consumidor de fluxo de dados

O `deregister-stream-consumer` exemplo a seguir cancela o registro do consumidor especificado do fluxo de dados especificado.

```
aws kinesis deregister-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

Este comando não produz saída.

Para obter mais informações, consulte [Desenvolvimento de consumidores com distribuição aprimorada usando o Kinesis Data Streams no Guia do desenvolvedor do Amazon Kinesis Data API Streams](#).

- Para API obter detalhes, consulte [DeregisterStreamConsumer](#) na Referência de AWS CLI Comandos.

describe-limits

O código de exemplo a seguir mostra como usar `describe-limits`.

AWS CLI

Para descrever os limites de fragmentos

O `describe-limits` exemplo a seguir mostra os limites de fragmentos e o uso da AWS conta atual.

```
aws kinesis describe-limits
```

Saída:

```
{
  "ShardLimit": 500,
  "OpenShardCount": 29
}
```

Para obter mais informações, consulte [Refragmentação de um stream no Guia](#) do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DescribeLimits](#) na Referência de AWS CLI Comandos.

describe-stream-consumer

O código de exemplo a seguir mostra como usar `describe-stream-consumer`.

AWS CLI

Para descrever um consumidor de fluxo de dados

O `describe-stream-consumer` exemplo a seguir retorna a descrição do consumidor especificado, registrado com o fluxo de dados especificado.

```
aws kinesis describe-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Leitura de dados do Amazon Kinesis Data Streams](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DescribeStreamConsumer](#) na Referência de AWS CLI Comandos.

describe-stream-summary

O código de exemplo a seguir mostra como usar `describe-stream-summary`.

AWS CLI

Para descrever um resumo do fluxo de dados

O `describe-stream-summary` exemplo a seguir fornece uma descrição resumida (sem a lista de fragmentos) do fluxo de dados especificado.

```
aws kinesis describe-stream-summary \  
  --stream-name samplestream
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Criar e gerenciar streamings](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DescribeStreamSummary](#) na Referência de AWS CLI Comandos.

describe-stream

O código de exemplo a seguir mostra como usar `describe-stream`.

AWS CLI

Como descrever um fluxo de dados

O exemplo de `describe-stream` a seguir retorna detalhes sobre o fluxo de dados especificado.

```
aws kinesis describe-stream \
  --stream-name samplestream
```

Saída:

```
{
  "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  
  }  
}
```

Para obter mais informações, consulte [Criar e gerenciar streamings](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DescribeStream](#) na Referência de AWS CLI Comandos.

disable-enhanced-monitoring

O código de exemplo a seguir mostra como usar `disable-enhanced-monitoring`.

AWS CLI

Para desativar o monitoramento aprimorado de métricas em nível de fragmento

O `disable-enhanced-monitoring` exemplo a seguir desativa o monitoramento aprimorado do stream de dados do Kinesis para métricas em nível de fragmento.

```
aws kinesis disable-enhanced-monitoring \  
  --stream-name samplestream --shard-level-metrics ALL
```

Saída:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ],  
  "DesiredShardLevelMetrics": []  
}
```

Para obter mais informações, consulte [Monitoramento de streams no Amazon Kinesis Data Streams](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [DisableEnhancedMonitoring](#) na Referência de AWS CLI Comandos.

enable-enhanced-monitoring

O código de exemplo a seguir mostra como usar `enable-enhanced-monitoring`.

AWS CLI

Para permitir o monitoramento aprimorado de métricas em nível de fragmento

O `enable-enhanced-monitoring` exemplo a seguir permite o monitoramento aprimorado do stream de dados do Kinesis para métricas em nível de fragmento.

```
aws kinesis enable-enhanced-monitoring \  
  --stream-name samplestream \  
  --shard-level-metrics ALL
```

Saída:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [],  
  "DesiredShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ]  
}
```

Para obter mais informações, consulte [Monitoramento de streams no Amazon Kinesis Data Streams](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [EnableEnhancedMonitoring](#) na Referência de AWS CLI Comandos.

get-records

O código de exemplo a seguir mostra como usar `get-records`.

AWS CLI

Como obter registros de um fragmento

O exemplo de `get-records` a seguir obtém registros de dados do fragmento de um fluxo de dados do Kinesis usando o iterador de fragmento especificado.

```
aws kinesis get-records \  
  --shard-iterator AAAAAAAAAAF7/0mWD7IuHj1yGv/  
TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYe0ZyLZrvzze0FVf9iN99hUPw/w/  
b0YWYeefNvnf1DYt5XpDJghLKr3DzgzknkTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/  
GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWGE1
```

Saída:

```
{  
  "Records": [],  
  "MillisBehindLatest": 80742000  
}
```

Para obter mais informações, consulte [Desenvolvimento de consumidores usando o Kinesis API Data Streams AWS SDK com o for Java no Guia do desenvolvedor do Amazon Kinesis Data Streams](#).

- Para API obter detalhes, consulte [GetRecords](#) na Referência de AWS CLI Comandos.

get-shard-iterator

O código de exemplo a seguir mostra como usar `get-shard-iterator`.

AWS CLI

Para obter um iterador de fragmentos

O `get-shard-iterator` exemplo a seguir usa o tipo iterador de `AT_SEQUENCE_NUMBER` fragmento e gera um iterador de fragmento para começar a ler os registros de dados exatamente da posição indicada pelo número de sequência especificado.

```
aws kinesis get-shard-iterator \  
  --stream-name samplestream \  
  --shard-id shardId-000000000001 \  
  --shard-iterator-type LATEST
```

Saída:

```
{  
  "ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqqH9FifJ+n48XWTh/  
IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy  
+lUkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn9408SqeEqwhigwZxNWxID3Ej7YYYcxQi8Q/fIrCjGAy/  
n2r5Z9G864YpWDFn9upNNQAR/ii0WKs"  
}
```

Para obter mais informações, consulte [Desenvolvimento de consumidores usando o Kinesis API Data Streams AWS SDK com o for Java no Guia do desenvolvedor do Amazon Kinesis Data Streams](#).

- Para API obter detalhes, consulte [GetShardIterator](#) na Referência de AWS CLI Comandos.

increase-stream-retention-period

O código de exemplo a seguir mostra como usar `increase-stream-retention-period`.

AWS CLI

Para aumentar o período de retenção do fluxo de dados

O `increase-stream-retention-period` exemplo a seguir aumenta o período de retenção (o período de tempo em que os registros de dados ficam acessíveis após serem adicionados ao fluxo) do fluxo especificado para 168 horas.

```
aws kinesis increase-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 168
```

Este comando não produz saída.

Para obter mais informações, consulte [Alteração do período de retenção de dados](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [IncreaseStreamRetentionPeriod](#) na Referência de AWS CLI Comandos.

list-shards

O código de exemplo a seguir mostra como usar `list-shards`.

AWS CLI

Para listar fragmentos em um fluxo de dados

O `list-shards` exemplo a seguir lista todos os fragmentos no fluxo especificado, começando com o fragmento cujo ID segue imediatamente o especificado `exclusive-start-shard-id` de `shardId-000000000000`

```
aws kinesis list-shards \  
  --stream-name samplestream \  
  --exclusive-start-shard-id shardId-000000000000
```

Saída:

```
{  
  "Shards": [  
    {  
      "ShardId": "shardId-000000000001",  
      "HashKeyRange": {  
        "StartingHashKey": "113427455640312821154458202477256070485",  
        "EndingHashKey": "226854911280625642308916404954512140969"  
      },  
      "SequenceNumberRange": {  
        "StartingSequenceNumber":  
"49600871682979337187563555549332609155523708941634633746"  
      }  
    },  
    {  
      "ShardId": "shardId-000000000002",  
      "HashKeyRange": {  
        "StartingHashKey": "226854911280625642308916404954512140970",  
        "EndingHashKey": "340282366920938463463374607431768211455"  
      },  
      "SequenceNumberRange": {
```

```
        "StartingSequenceNumber":  
        "49600871683001637932762086172474144873796357303140614178"  
    }  
  }  
]  
}
```

Para obter mais informações, consulte [Listing Shards](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [ListShards](#) na Referência de AWS CLI Comandos.

list-streams

O código de exemplo a seguir mostra como usar `list-streams`.

AWS CLI

Como listar fluxos de dados

O exemplo de `list-streams` a seguir lista todos os fluxos de dados ativos na conta e região atuais.

```
aws kinesis list-streams
```

Saída:

```
{  
  "StreamNames": [  
    "samplestream",  
    "samplestream1"  
  ]  
}
```

Para obter mais informações, consulte [Listar streams](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [ListStreams](#) na Referência de AWS CLI Comandos.

list-tags-for-stream

O código de exemplo a seguir mostra como usar `list-tags-for-stream`.

AWS CLI

Para listar tags para um fluxo de dados

O `list-tags-for-stream` exemplo a seguir lista as tags anexadas ao fluxo de dados especificado.

```
aws kinesis list-tags-for-stream \  
  --stream-name samplestream
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "samplekey",  
      "Value": "example"  
    }  
  ],  
  "HasMoreTags": false  
}
```

Para obter mais informações, consulte Como [marcar seus streams no Guia](#) do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [ListTagsForStream](#) na Referência de AWS CLI Comandos.

merge-shards

O código de exemplo a seguir mostra como usar `merge-shards`.

AWS CLI

Para mesclar fragmentos

O `merge-shards` exemplo a seguir mescla dois fragmentos adjacentes com IDs `shardId-000000000000` e `shardId-000000000001` no fluxo de dados especificado e os combina em um único fragmento.

```
aws kinesis merge-shards \  
  --stream-name samplestream \  
  --shard-to-merge shardId-000000000000 \  
  --shard-to-merge shardId-000000000001
```

```
--adjacent-shard-to-merge shardId-000000000001
```

Este comando não produz saída.

Para obter mais informações, consulte [Mesclando dois fragmentos no Guia](#) do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [MergeShards](#) na Referência de AWS CLI Comandos.

put-record

O código de exemplo a seguir mostra como usar `put-record`.

AWS CLI

Como gravar um registro em um fluxo de dados

O exemplo de `put-record` a seguir grava um único registro de dados no fluxo de dados especificado usando a chave de partição especificada.

```
aws kinesis put-record \  
  --stream-name samplestream \  
  --data sampledatarecord \  
  --partition-key samplepartitionkey
```

Saída:

```
{  
  "ShardId": "shardId-0000000000009",  
  "SequenceNumber": "49600902273357540915989931256901506243878407835297513618",  
  "EncryptionType": "KMS"  
}
```

Para obter mais informações, consulte [Desenvolvendo produtores usando o Amazon Kinesis API Data Streams AWS SDK com o for Java no Guia do desenvolvedor do](#) Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [PutRecord](#) na Referência de AWS CLI Comandos.

put-records

O código de exemplo a seguir mostra como usar `put-records`.

AWS CLI

Para gravar vários registros em um fluxo de dados

O `put-records` exemplo a seguir grava um registro de dados usando a chave de partição especificada e outro registro de dados usando uma chave de partição diferente em uma única chamada.

```
aws kinesism put-records \  
  --stream-name samplestream \  
  --  
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2
```

Saída:

```
{  
  "FailedRecordCount": 0,  
  "Records": [  
    {  
      "SequenceNumber":  
"49600883331171471519674795588238531498465399900093808706",  
      "ShardId": "shardId-000000000004"  
    },  
    {  
      "SequenceNumber":  
"49600902273357540915989931256902715169698037101720764562",  
      "ShardId": "shardId-000000000009"  
    }  
  ],  
  "EncryptionType": "KMS"  
}
```

Para obter mais informações, consulte [Desenvolvendo produtores usando o Amazon Kinesis API Data Streams AWS SDK com o for Java no Guia do desenvolvedor do Amazon Kinesis Data Streams](#).

- Para API obter detalhes, consulte [PutRecords](#) na Referência de AWS CLI Comandos.

register-stream-consumer

O código de exemplo a seguir mostra como usar `register-stream-consumer`.

AWS CLI

Para registrar um consumidor de fluxo de dados

O `register-stream-consumer` exemplo a seguir registra um consumidor chamado `KinesisConsumerApplication` com o fluxo de dados especificado.

```
aws kinesis register-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

Saída:

```
{  
  "Consumer": {  
    "ConsumerName": "KinesisConsumerApplication",  
    "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/  
consumer/KinesisConsumerApplication:1572383852",  
    "ConsumerStatus": "CREATING",  
    "ConsumerCreationTimestamp": 1572383852.0  
  }  
}
```

Para obter mais informações, consulte [Desenvolvimento de consumidores com distribuição aprimorada usando o Kinesis Data Streams no Guia do desenvolvedor do Amazon Kinesis Data API Streams](#).

- Para API obter detalhes, consulte [RegisterStreamConsumer](#) na Referência de AWS CLI Comandos.

remove-tags-from-stream

O código de exemplo a seguir mostra como usar `remove-tags-from-stream`.

AWS CLI

Para remover tags de um fluxo de dados

O `remove-tags-from-stream` exemplo a seguir remove a tag com a chave especificada do fluxo de dados especificado.

```
aws kinesis remove-tags-from-stream \  
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \  
  --tag-key my-tag-key
```



```
--stream-name samplestream \  
--tag-keys samplekey
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar seus streams no Guia](#) do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [RemoveTagsFromStream](#) na Referência de AWS CLI Comandos.

split-shard

O código de exemplo a seguir mostra como usar `split-shard`.

AWS CLI

Para dividir fragmentos

O `split-shard` exemplo a seguir divide o fragmento especificado em dois novos fragmentos usando uma nova chave de hash inicial de 10.

```
aws kinesis split-shard \  
  --stream-name samplestream \  
  --shard-to-split shardId-000000000000 \  
  --new-starting-hash-key 10
```

Este comando não produz saída.

Para obter mais informações, consulte [Dividindo um fragmento](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [SplitShard](#) na Referência de AWS CLI Comandos.

start-stream-encryption

O código de exemplo a seguir mostra como usar `start-stream-encryption`.

AWS CLI

Para habilitar a criptografia de fluxo de dados

O `start-stream-encryption` exemplo a seguir ativa a criptografia do lado do servidor para o fluxo especificado, usando a chave especificada. 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Proteção de dados no Amazon Kinesis Data Streams](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [StartStreamEncryption](#) na Referência de AWS CLI Comandos.

stop-stream-encryption

O código de exemplo a seguir mostra como usar `stop-stream-encryption`.

AWS CLI

Para desativar a criptografia do fluxo de dados

O `stop-stream-encryption` exemplo a seguir desativa a criptografia do lado do servidor para o fluxo especificado, usando a chave especificada. 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Proteção de dados no Amazon Kinesis Data Streams](#) no Guia do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [StopStreamEncryption](#) na Referência de AWS CLI Comandos.

update-shard-count

O código de exemplo a seguir mostra como usar `update-shard-count`.

AWS CLI

Para atualizar a contagem de fragmentos em um fluxo de dados

O `update-shard-count` exemplo a seguir atualiza a contagem de fragmentos do fluxo de dados especificado para 6. Este exemplo usa escala uniforme, que cria fragmentos do mesmo tamanho.

```
aws kinesis update-shard-count \  
  --stream-name samplestream \  
  --scaling-type UNIFORM_SCALING \  
  --target-shard-count 6
```

Saída:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardCount": 3,  
  "TargetShardCount": 6  
}
```

Para obter mais informações, consulte [Refragmentação de um stream no Guia](#) do desenvolvedor do Amazon Kinesis Data Streams.

- Para API obter detalhes, consulte [UpdateShardCount](#) na Referência de AWS CLI Comandos.

AWS KMS exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS KMS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-key-deletion

O código de exemplo a seguir mostra como usar `cancel-key-deletion`.

AWS CLI

Para cancelar a exclusão programada de uma chave gerenciada pelo KMS cliente

O `cancel-key-deletion` exemplo a seguir cancela a exclusão programada de uma chave gerenciada pelo KMS cliente.

```
aws kms cancel-key-deletion \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Quando o `cancel-key-deletion` comando for bem-sucedido, a exclusão programada será cancelada. No entanto, o estado chave da KMS chave é `Disabled` que você não pode usar a KMS chave em operações criptográficas. Para restaurar sua funcionalidade, use o `enable-key` comando.

Para obter mais informações, consulte [Agendamento e cancelamento da exclusão de chaves no Guia](#) do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [CancelKeyDeletion](#) na Referência de AWS CLI Comandos.

connect-custom-key-store

O código de exemplo a seguir mostra como usar `connect-custom-key-store`.

AWS CLI

Para conectar um armazenamento de chaves personalizado

O `connect-custom-key-store` exemplo a seguir reconecta o armazenamento de chaves personalizadas especificado. Você pode usar um comando como esse para conectar um armazenamento de chaves personalizado pela primeira vez ou para reconectar um armazenamento de chaves que foi desconectado.

Você pode usar esse comando para conectar um armazenamento de HSM chaves do AWS Cloud ou um armazenamento de chaves externo.

```
aws kms connect-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Esse comando não retorna nenhuma saída. Use o comando `describe-custom-key-stores` para verificar se o comando está em vigor.

Para obter informações sobre como conectar um armazenamento de HSM chaves do AWS Cloud, consulte [Como conectar e desconectar um armazenamento de HSM chaves do AWS Cloud](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter informações sobre como conectar um armazenamento de chaves externo, consulte [Conectando e desconectando um armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ConnectCustomKeyStore](#) na Referência de AWS CLI Comandos.

create-alias

O código de exemplo a seguir mostra como usar `create-alias`.

AWS CLI

Para criar um alias para uma chave KMS

O `create-alias` comando a seguir cria um alias com o nome `example-alias` da KMS chave identificada pelo ID `1234abcd-12ab-34cd-56ef-1234567890ab` da chave.

Os nomes de alias devem começar com `alias/`. Não use nomes de alias que comecem com `alias/aws`; eles são reservados para uso por AWS.

```
aws kms create-alias \  
  --alias-name alias/example-alias \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Esse comando não retorna resultados. Use o `list-aliases` comando para visualizar o novo alias.

Para obter mais informações, consulte [Using aliases](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [CreateAlias](#) na Referência de AWS CLI Comandos.

create-custom-key-store

O código de exemplo a seguir mostra como usar `create-custom-key-store`.

AWS CLI

Exemplo 1: Para criar um armazenamento de HSM chaves AWS na nuvem

O `create-custom-key-store` exemplo a seguir cria um armazenamento de HSM chaves do AWS Cloud apoiado por um HSM cluster do AWS Cloud usando os parâmetros necessários. Você também pode adicionar `custom-key-store-type` parameter with the default value: `AWS_CLOUDHSM`.

Para especificar a entrada do arquivo para o `trust-anchor-certificate` comando no AWS CLI, o `file://` prefixo é necessário.

```
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
```

Saída:

```
{  
  "CustomKeyId": cks-1234567890abcdef0  
}
```

Para obter mais informações, consulte [Como criar um armazenamento de HSM chaves na AWS nuvem](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: Para criar um armazenamento de chaves externo com conectividade de endpoint público

O `create-custom-key-store` exemplo a seguir cria um armazenamento de chaves externo (XKS) que se comunica pela AWS KMS Internet.

Neste exemplo, o `XksProxyUriPath` usa um prefixo opcional `deexample-prefix`.

NOTE: se você usa a AWS CLI versão 1.0, execute o comando a seguir antes de especificar um parâmetro com um HTTPS valor HTTP ou, como o `XksProxyUriEndpoint` parâmetro.

```
aws configure set cli_follow_urlparam false
```

Caso contrário, a AWS CLI versão 1.0 substitui o valor do parâmetro pelo conteúdo encontrado nesse URI endereço.

```
aws kms create-custom-key-store \  
  --custom-key-store-name ExamplePublicEndpointXKS \  
  --custom-key-store-type EXTERNAL_KEY_STORE \  
  --xks-proxy-connectivity PUBLIC_ENDPOINT \  
  --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \  
  --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \  
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Saída:

```
{  
  "CustomKeyStoreId": cks-2234567890abcdef0  
}
```

Para obter mais informações, consulte [Criação de um armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 3: Para criar um armazenamento de chaves externo com conectividade de serviço de VPC endpoint

O `create-custom-key-store` exemplo a seguir cria um armazenamento de chaves externo (XKS) que usa um serviço de VPC endpoint da Amazon com AWS KMS o qual se comunicar.

NOTE: se você usa a AWS CLI versão 1.0, execute o comando a seguir antes de especificar um parâmetro com um HTTPS valor HTTP ou, como o `XksProxyUriEndpoint` parâmetro.

```
aws configure set cli_follow_urlparam false
```

Caso contrário, a AWS CLI versão 1.0 substitui o valor do parâmetro pelo conteúdo encontrado nesse URI endereço.

```
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="
```

Saída:

```
{  
  "CustomKeyStoreId": cks-3234567890abcdef0  
}
```

Para obter mais informações, consulte [Criação de um armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [CreateCustomKeyStore](#) na Referência de AWS CLI Comandos.

create-grant

O código de exemplo a seguir mostra como usar `create-grant`.

AWS CLI

Como criar uma concessão

O `create-grant` exemplo a seguir cria uma concessão que permite ao `exampleUser` usuário usar o `decrypt` comando na KMS chave de `1234abcd-12ab-34cd-56ef-1234567890ab` exemplo. A entidade principal descontinuada é o perfil `adminRole`. A concessão usa a restrição de concessão `EncryptionContextSubset` para conceder essa permissão apenas quando o contexto de criptografia na solicitação `decrypt` incluir o par de chave/valor `"Department": "IT"`.

```
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
```

Saída:

```
{  
  "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",  
  "GrantToken": "<grant token here>"  
}
```

Use o `list-grants` comando para visualizar informações detalhadas sobre a concessão.

Para obter mais informações, consulte [Concessões AWS KMS no](#) Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [CreateGrant](#) na Referência de AWS CLI Comandos.

create-key

O código de exemplo a seguir mostra como usar `create-key`.

AWS CLI

Exemplo 1: Para criar uma KMS chave gerenciada pelo cliente no AWS KMS

O `create-key` exemplo a seguir cria uma KMS chave de criptografia simétrica.

Para criar a KMS chave básica, uma chave de criptografia simétrica, você não precisa especificar nenhum parâmetro. Os valores padrão desses parâmetros criam uma chave de criptografia simétrica.

Como esse comando não especifica uma política de chaves, a KMS chave obtém a [política de chaves padrão](#) para chaves criadas KMS programaticamente. Use o comando `get-key-policy` para visualizar a política de chave. Use o comando `put-key-policy` para alterar a política de chave.

```
aws kms create-key
```

O `create-key` comando retorna os metadados da chave, incluindo o ID da chave e ARN da nova KMS chave. Você pode usar esses valores para identificar a KMS chave em outras AWS KMS operações. A saída não inclui as tags. Para ver as tags de uma KMS chave, use `list-resource-tags` command o.

Saída:

```
{
  "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: O `create-key` comando não permite que você especifique um alias. Para criar um alias para a nova KMS chave, use o `create-alias` comando.

Para obter mais informações, consulte [Creating keys](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: Para criar uma RSA KMS chave assimétrica para criptografia e decodificação

O `create-key` exemplo a seguir cria uma KMS chave que contém um par de RSA chaves assimétrico para criptografia e decodificação.

```
aws kms create-key \  
  --key-spec RSA_4096 \  
  --key-usage ENCRYPT_DECRYPT
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Chaves assimétricas AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 3: Para criar uma KMS chave de curva elíptica assimétrica para assinatura e verificação

Para criar uma KMS chave assimétrica que contenha um ECC par de chaves curva elíptica assimétrica () para assinatura e verificação. O `--key-usage` parâmetro é obrigatório, embora `SIGN_VERIFY` seja o único valor válido para ECC KMS chaves.

```
aws kms create-key \  
  --key-spec ECC_NIST_P521 \  
  --key-usage SIGN_VERIFY
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

Para obter mais informações, consulte [Chaves assimétricas AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 4: Para criar uma HMAC KMS chave

O create-key exemplo a seguir cria uma chave de 384 bits HMACKMS. O GENERATE_VERIFY_MAC valor do --key-usage parâmetro é obrigatório, embora seja o único valor válido para HMAC KMS chaves.

```
aws kms create-key \  
  --key-spec HMAC_384 \  
  --key-usage GENERATE_VERIFY_MAC
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [HMACAs chaves AWS KMS no](#) Guia do desenvolvedor do AWS Key Management Service.

Exemplo 4: Para criar uma chave primária KMS multirregional

O exemplo de `create-key` a seguir cria uma chave primária de criptografia simétrica multirregional. Como os valores padrão para todos os parâmetros criam uma chave de criptografia simétrica, somente o `--multi-region` parâmetro é necessário para essa KMS chave. No AWS CLI, para indicar que um parâmetro booleano é verdadeiro, basta especificar o nome do parâmetro.

```
aws kms create-key \
  --multi-region
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Chaves assimétricas AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 5: Para criar uma KMS chave para material de chave importado

O `create-key` exemplo a seguir cria uma KMS chave sem material de chave. Quando a operação estiver concluída, você poderá importar seu próprio material de chave para a KMS chave. Para criar essa KMS chave, defina o `--origin` parâmetro como `EXTERNAL`.

```

aws kms create-key \
  --origin EXTERNAL

```

Saída:

```
{
```

```

    "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"
    }
  }
}

```

Para obter mais informações, consulte [Importação de material de AWS KMS chaves em chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 6: Para criar uma KMS chave em um armazenamento de HSM chaves do AWS Cloud

O `create-key` exemplo a seguir cria uma KMS chave no armazenamento de HSM chaves do AWS Cloud especificado. A operação cria a KMS chave e seus metadados AWS KMS e cria o material da chave no HSM cluster AWS Cloud associado ao armazenamento de chaves personalizadas. Os parâmetros `--custom-key-store-id` e `--origin` são obrigatórios.

```

aws kms create-key \
  --origin AWS_CLOUDHSM \
  --custom-key-store-id cks-1234567890abcdef0

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Armazenamentos de HSM chaves na AWS nuvem](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 7: Para criar uma KMS chave em um armazenamento de chaves externo

O `create-key` exemplo a seguir cria uma KMS chave no armazenamento de chaves externo especificado. Os parâmetros `--custom-key-store-id`, `--origin` e `--xks-key-id` são obrigatórios neste comando.

O parâmetro `--xks-key-id` especifica o ID de uma chave de criptografia simétrica existente no gerenciador de chaves externo. Essa chave serve como material de chave externa para a KMS chave. O valor do `--origin` parâmetro deve ser `EXTERNAL_KEY_STORE`. O `custom-key-store-id` parâmetro deve identificar um armazenamento de chaves externo conectado ao proxy externo do armazenamento de chaves.

```

aws kms create-key \
  --origin EXTERNAL_KEY_STORE \
  --custom-key-store-id cks-9876543210fedcba9 \
  --xks-key-id bb8562717f809024

```

Saída:

```
{
```



```
"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"
  }
}
```

Para obter mais informações, consulte [External key stores](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [CreateKey](#) na Referência de AWS CLI Comandos.

decrypt

O código de exemplo a seguir mostra como usar `decrypt`.

AWS CLI

Exemplo 1: Para descriptografar uma mensagem criptografada com uma chave simétrica (Linux e KMS macOS)

O exemplo de `decrypt` comando a seguir demonstra a maneira recomendada de descriptografar dados com o AWS CLI. Esta versão mostra como descriptografar dados com uma chave simétrica. KMS

Forneça o texto cifrado em um arquivo. No valor do `--ciphertext-blob` parâmetro, use o `fileb://` prefixo, que instrui o CLI a ler os dados de um arquivo binário. Se o arquivo não estiver no diretório atual, digite o caminho completo para o arquivo. Para obter mais informações sobre a leitura de valores de AWS CLI parâmetros de um arquivo, consulte Carregando AWS CLI parâmetros de um arquivo < <https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html> > no Guia do usuário da interface de linha de AWS comando e práticas recomendadas para parâmetros de arquivos locais < <https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/> > no blog da ferramenta de linha de comando. Especifique a chave para descriptografar o texto cifrado. O parâmetro não é necessário ao descriptografar com uma chave AWS simétrica. KMS `--key-id` KMS AWS KMS pode obter o ID da KMS chave que foi usada para criptografar os dados dos metadados no texto cifrado. Mas é sempre uma boa prática especificar a KMS chave que você está usando. Essa prática garante que você use a KMS chave desejada e evita que você decifre inadvertidamente um texto cifrado usando uma KMS chave na qual você não confia. Solicite a saída de texto sem formatação como um valor de texto. O parâmetro `--query` CLI Plaintext O parâmetro `--output` retorna a saída como texto. Decodifique o texto simples em Base64 e salve-o em um arquivo. O seguinte exemplo canaliza (|) o valor do parâmetro Plaintext para o utilitário Base64, que o decodifica. Em seguida, ele redireciona (>) a saída decodificada para o arquivo ExamplePlaintextFile.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms decrypt \  
  --ciphertext-blob fileb://ExampleEncryptedFile \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query Plaintext | base64 \  
  --decode > ExamplePlaintextFile
```

Este comando não produz saída. A saída do comando `decrypt` é decodificada em base64 e salva em um arquivo.

Para obter mais informações, consulte [Decrypt](#) na Referência do Serviço de Gerenciamento de AWS Chaves. API

Exemplo 2: Para descriptografar uma mensagem criptografada com uma KMS chave simétrica (prompt de comando do Windows)

O exemplo a seguir é igual ao anterior, mas ele usa o utilitário `certutil` para decodificar em Base64 os dados em texto simples. Esse procedimento requer dois comandos, conforme mostrado nos exemplos a seguir.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms decrypt ^
  --ciphertext-blob fileb://ExampleEncryptedFile ^
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
  --output text ^
  --query Plaintext > ExamplePlaintextFile.base64
```

Execute o comando `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Saída:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Para obter mais informações, consulte [Decrypt](#) na Referência do Serviço de Gerenciamento de AWS Chaves. API

Exemplo 3: Para descriptografar uma mensagem criptografada com uma chave assimétrica KMS (Linux e macOS)

O exemplo de `decrypt` comando a seguir mostra como descriptografar dados criptografados sob uma RSA chave assimétrica. KMS

Ao usar uma KMS chave assimétrica, o `encryption-algorithm` parâmetro, que especifica o algoritmo usado para criptografar o texto sem formatação, é obrigatório.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms decrypt \  
  --ciphertext-blob fileb://ExampleEncryptedFile \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query Plaintext > ExamplePlaintextFile.base64
```

```
--key-id 0987dcb-a-09fe-87dc-65ba-ab0987654321 \  
--encryption-algorithm RSAES_OAEP_SHA_256 \  
--output text \  
--query Plaintext | base64 \  
--decode > ExamplePlaintextFile
```

Este comando não produz saída. A saída do comando `decrypt` é decodificada em base64 e salva em um arquivo.

Para obter mais informações, consulte [Chaves assimétricas AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [Decrypt](#) na AWS CLI Referência de Comandos.

delete-alias

O código de exemplo a seguir mostra como usar `delete-alias`.

AWS CLI

Para excluir um AWS KMS alias

O exemplo de `delete-alias` a seguir exclui o alias `alias/example-alias`. O nome de alias deve começar com `alias/`.

```
aws kms delete-alias \  
--alias-name alias/example-alias
```

Este comando não produz saída. Use o `list-aliases` comando para localizar o alias.

Para obter mais informações, consulte [Deleting an alias](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [DeleteAlias](#) na Referência de AWS CLI Comandos.

delete-custom-key-store

O código de exemplo a seguir mostra como usar `delete-custom-key-store`.

AWS CLI

Para excluir um armazenamento de chaves personalizado

O `delete-custom-key-store` exemplo a seguir exclui o armazenamento de chaves personalizadas especificado.

A exclusão de um armazenamento de HSM chaves do AWS Cloud não tem efeito no HSM cluster do Cloud associado. A exclusão de um armazenamento de chaves externo não afeta o proxy do armazenamento de chaves externo associado, o gerenciador de chaves externo ou as chaves externas.

NOTE: antes de excluir um armazenamento de chaves personalizado, você deve agendar a exclusão de todas KMS as chaves no armazenamento de chaves personalizadas e esperar que essas KMS chaves sejam excluídas. Em seguida, você deve desconectar o armazenamento de chaves personalizadas. Para obter ajuda para encontrar as KMS chaves em seu armazenamento de chaves personalizado, consulte [Excluir um armazenamento de HSM chaves na AWS nuvem \(API\)](#) no Guia do desenvolvedor do AWS Key Management Service.

```
delete-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Esse comando não retorna nenhuma saída. Para verificar se o armazenamento de chaves personalizadas foi excluído, use o `describe-custom-key-stores` comando.

Para obter informações sobre como excluir um AWS armazenamento de HSM chaves do Cloud, consulte [Excluindo um armazenamento de HSM chaves do AWS Cloud](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter informações sobre como excluir armazenamentos de chaves externos, consulte [Excluindo um armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [DeleteCustomKeyStore](#) na Referência de AWS CLI Comandos.

delete-imported-key-material

O código de exemplo a seguir mostra como usar `delete-imported-key-material`.

AWS CLI

Para excluir material de chave importado de uma KMS chave

O `delete-imported-key-material` exemplo a seguir exclui o material de chave que foi importado para uma KMS chave.

```
aws kms delete-imported-key-material \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Para verificar se o material da chave foi excluído, use o `describe-key` comando para procurar um estado de chave de `PendingImport` ou `PendingDeletion`.

Para obter mais informações, consulte [Excluindo o material da chave importada](https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key) < <https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key> do Key Management Service.AWS

- Para API obter detalhes, consulte [DeleteImportedKeyMaterial](#) na Referência de AWS CLI Comandos.

describe-custom-key-stores

O código de exemplo a seguir mostra como usar `describe-custom-key-stores`.

AWS CLI

Exemplo 1: Para obter detalhes sobre um armazenamento de HSM chaves do AWS Cloud

O `describe-custom-key-store` exemplo a seguir exibe detalhes sobre o armazenamento de HSM chaves do AWS Cloud especificado. O comando é o mesmo para todos os tipos de armazenamento de chaves personalizadas, mas a saída difere com o tipo de armazenamento de chaves e, para um armazenamento de chaves externo, com sua opção de conectividade.

Por padrão, esse comando exibe informações sobre todos os armazenamentos de chaves personalizadas na conta e na região. Para exibir informações sobre um determinado armazenamento de chaves personalizadas, use o `custom-key-store-id` parâmetro `custom-key-store-name` ou.

```
aws kms describe-custom-key-stores \  
  --custom-key-store-name ExampleCloudHSMKeyStore
```

A saída desse comando inclui detalhes úteis sobre o armazenamento de HSM chaves do AWS Cloud, incluindo seu estado de conexão (`ConnectionState`). Se o estado da conexão for `FAILED`, a saída incluirá um `ConnectionErrorCode` campo que descreve o problema.

Saída:

```
{
  "CustomKeyStores": [
    {
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-04-05T14:04:55-07:00",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleExternalKeyStore",
      "TrustAnchorCertificate": "<certificate appears here>"
    }
  ]
}
```

Para obter mais informações, consulte [Como visualizar um armazenamento de HSM chaves na AWS nuvem](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: Para obter detalhes sobre um armazenamento de chaves externo com conectividade de endpoint público

O `describe-custom-key-store` exemplo a seguir exibe detalhes sobre o armazenamento de chaves externo especificado. O comando é o mesmo para todos os tipos de armazenamento de chaves personalizadas, mas a saída difere com o tipo de armazenamento de chaves e, para um armazenamento de chaves externo, com sua opção de conectividade.

Por padrão, esse comando exibe informações sobre todos os armazenamentos de chaves personalizadas na conta e na região. Para exibir informações sobre um determinado armazenamento de chaves personalizadas, use o `custom-key-store-id` parâmetro `custom-key-store-name` ou.

```
aws kms describe-custom-key-stores \
  --custom-key-store-id cks-9876543210fedcba9
```

A saída desse comando inclui detalhes úteis sobre o armazenamento externo de chaves, incluindo seu estado de conexão (`ConnectionState`). Se o estado da conexão for `FAILED`, a saída incluirá um `ConnectionErrorCode` campo que descreve o problema.

Saída:

```
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-9876543210fedcba9",
      "CustomKeyName": "ExampleXKS",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-02T07:48:55-07:00",
      "CustomKeyType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://myproxy.xks.example.com",
        "UriPath": "/example-prefix/kms/xks/v1"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Visualização de um armazenamento externo de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 3: Para obter detalhes sobre um armazenamento de chaves externo com conectividade de serviço de VPC endpoint

O `describe-custom-key-store` exemplo a seguir exibe detalhes sobre o armazenamento de chaves externo especificado. O comando é o mesmo para todos os tipos de armazenamento de chaves personalizadas, mas a saída difere com o tipo de armazenamento de chaves e, para um armazenamento de chaves externo, com sua opção de conectividade.

Por padrão, esse comando exibe informações sobre todos os armazenamentos de chaves personalizadas na conta e na região. Para exibir informações sobre um determinado armazenamento de chaves personalizadas, use o `custom-key-store-id` parâmetro `custom-key-store-name` ou.

```
aws kms describe-custom-key-stores \
  --custom-key-store-id cks-2234567890abcdef0
```


A saída desse comando inclui detalhes úteis sobre o armazenamento externo de chaves, incluindo seu estado de conexão (`ConnectionState`). Se o estado da conexão for `FAILED`, a saída incluirá um `ConnectionErrorCode` campo que descreve o problema.

Saída:

```
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-3234567890abcdef0",
      "CustomKeyName": "ExampleVPCEExternalKeyStore",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-22T07:48:55-07:00",
      "CustomKeyType": "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Visualização de um armazenamento externo de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [DescribeCustomKeyStores](#) na Referência de AWS CLI Comandos.

describe-key

O código de exemplo a seguir mostra como usar `describe-key`.

AWS CLI

Exemplo 1: Para encontrar informações detalhadas sobre uma KMS chave

O `describe-key` exemplo a seguir obtém informações detalhadas sobre a chave AWS gerenciada para o Amazon S3 na conta e região do exemplo. Você pode usar esse comando para encontrar detalhes sobre chaves AWS gerenciadas e chaves gerenciadas pelo cliente.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um valor de nome de alias, mas você pode usar um ID de chave, chaveARN, nome de alias ou alias ARN nesse comando.

```
aws kms describe-key \  
  --key-id alias/aws/s3
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

Para obter mais informações, consulte [Viewing keys](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: Para obter detalhes sobre uma chave RSA assimétrica KMS

O `describe-key` exemplo a seguir obtém informações detalhadas sobre uma RSA KMS chave assimétrica usada para assinatura e verificação.

```
aws kms describe-key \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

Exemplo 3: como obter detalhes sobre uma chave de réplica multirregional

O exemplo de `describe-key` a seguir os obtém metadados de uma chave de réplica multirregional. Essa chave multirregional é uma chave de criptografia simétrica. A saída de um comando `describe-key` para qualquer chave multirregional retorna informações sobre a chave primária e todas as suas réplicas.

```
aws kms describe-key \  

```

```
--key-id arn:aws:kms:ap-northeast-1:111122223333:key/  
mrk-1234abcd12ab34cd56ef1234567890ab
```

Saída:

```
{  
  "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"
    }
]
}
}
}
}

```

Exemplo 4: Para obter detalhes sobre uma HMAC KMS chave

O `describe-key` exemplo a seguir obtém informações detalhadas sobre uma HMAC KMS chave.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Saída:

```

{
  "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
  }
}

```

- Para API obter detalhes, consulte [DescribeKey](#) na Referência de AWS CLI Comandos.

disable-key-rotation

O código de exemplo a seguir mostra como usar `disable-key-rotation`.

AWS CLI

Para desativar a rotação automática de uma KMS chave

O `disable-key-rotation` exemplo a seguir desativa a rotação automática de uma KMS chave gerenciada pelo cliente. Para reativar a rotação automática, use o `enable-key-rotation` comando.

```
aws kms disable-key-rotation \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Para verificar se a rotação automática está desativada para a KMS chave, use o `get-key-rotation-status` comando.

Para obter mais informações, consulte [Chaves rotativas](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [DisableKeyRotation](#) na Referência de AWS CLI Comandos.

disable-key

O código de exemplo a seguir mostra como usar `disable-key`.

AWS CLI

Para desativar temporariamente uma KMS chave

O exemplo a seguir usa o `disable-key` comando para desativar uma KMS chave gerenciada pelo cliente. Para reativar a KMS chave, use o `enable-key` comando.

```
aws kms disable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída.

Para obter mais informações, consulte [Enabling and Disabling Keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [DisableKey](#) na Referência de AWS CLI Comandos.

disconnect-custom-key-store

O código de exemplo a seguir mostra como usar `disconnect-custom-key-store`.

AWS CLI

Para desconectar um armazenamento de chaves personalizadas

O `disconnect-custom-key-store` exemplo a seguir desconecta um armazenamento de chaves personalizadas do HSM cluster AWS Cloud. Você pode desconectar um armazenamento de chaves para solucionar um problema, atualizar suas configurações ou impedir que as chaves do repositório de KMS chaves sejam usadas em operações criptográficas.

Esse comando é o mesmo para todos os armazenamentos de chaves personalizados, incluindo armazenamentos de HSM chaves AWS na nuvem e armazenamentos de chaves externos.

Antes de executar um comando como esse, substitua o ID de exemplo do armazenamento de chaves personalizado por um válido.

```
$ aws kms disconnect-custom-key-store \
  --custom-key-store-id cks-1234567890abcdef0
```

Esse comando não produz saída. Verifique se o comando foi efetivo, use o `describe-custom-key-stores` comando.

Para obter mais informações sobre como desconectar um armazenamento de HSM chaves do AWS Cloud, consulte Como [conectar e desconectar um armazenamento de HSM chaves do AWS Cloud](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter mais informações sobre como desconectar um armazenamento de chaves externo, consulte [Conectando e desconectando um armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [DisconnectCustomKeyStore](#) na Referência de AWS CLI Comandos.

enable-key-rotation

O código de exemplo a seguir mostra como usar `enable-key-rotation`.

AWS CLI

Para ativar a rotação automática de uma KMS chave

O `enable-key-rotation` exemplo a seguir permite a rotação automática de uma KMS chave gerenciada pelo cliente com um período de rotação de 180 dias. A KMS chave será rotacionada um ano (aproximadamente 365 dias) a partir da data em que esse comando for concluído e todos os anos a partir de então.

O `--key-id` parâmetro identifica a KMS chave. Este exemplo usa um ARN valor de chave, mas você pode usar o ID da chave ou o ARN da KMS chave. O `--rotation-period-in-days` parâmetro especifica o número de dias entre cada data de rotação. Especifique um valor entre 90 e 2560 dias. Se nenhum valor for especificado, o valor padrão será 365 dias.

```
aws kms enable-key-rotation \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
  --rotation-period-in-days 180
```

Este comando não produz saída. Para verificar se a KMS chave está ativada, use o `get-key-rotation-status` comando.

Para obter mais informações, consulte [Chaves rotativas](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [EnableKeyRotation](#) na Referência de AWS CLI Comandos.

enable-key

O código de exemplo a seguir mostra como usar `enable-key`.

AWS CLI

Para habilitar uma KMS chave

O exemplo de `enable-key` a seguir habilita uma chave gerenciada pelo cliente. Você pode usar um comando como esse para ativar uma KMS chave que você desativou temporariamente usando o `disable-key` comando. Você também pode usá-lo para ativar uma KMS chave que está desativada porque foi programada para exclusão e a exclusão foi cancelada.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou ARN valor de chave neste comando.

Antes de executar esse comando, substitua o ID da chave de exemplo por um ID válido.

```
aws kms enable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Para verificar se a KMS chave está ativada, use o `describe-key` comando. Consulte os valores dos campos `KeyState` e `Enabled` na saída `describe-key`.

Para obter mais informações, consulte [Enabling and Disabling Keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [EnableKey](#) na Referência de AWS CLI Comandos.

encrypt

O código de exemplo a seguir mostra como usar `encrypt`.

AWS CLI

Exemplo 1: como criptografar o conteúdo de um arquivo no Linux ou no macOS

O `encrypt` comando a seguir demonstra a forma recomendada de criptografar dados com o AWS CLI

```
aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

O comando realiza diversas ações:

Usa o parâmetro `--plaintext` para indicar os dados a serem criptografados. Esse valor de parâmetro deve ser codificado em Base64. O valor do `plaintext` parâmetro deve ser codificado em base64 ou você deve usar o `fileb://` prefixo, que diz AWS CLI ao para ler dados binários do arquivo. Se o arquivo não estiver no diretório atual, digite o caminho completo para o arquivo. Por exemplo: `fileb:///var/tmp/ExamplePlaintextFile` ou `fileb://C:\Temp\ExamplePlaintextFile`. [Para obter mais informações sobre a leitura de valores de AWS CLI parâmetros de um arquivo, consulte Carregando parâmetros de um arquivo](#)

[no Guia do usuário da interface de linha de AWS comando e práticas recomendadas para parâmetros de arquivos locais no blog da ferramenta de linha de AWS comando. Usa `--query` os parâmetros `--output` e para controlar a saída do comando. Esses parâmetros extraem os dados criptografados, chamados de texto cifrado, da saída do comando. Para obter mais informações sobre como controlar a saída, consulte \[Controle de saída Saída de comando\]\(#\) no Guia do usuário da interface de linha de AWS comando. Usa o `base64` utilitário para decodificar a saída extraída em dados binários. O texto cifrado retornado por um comando bem-sucedido é um texto codificado em `base64`. `encrypt` Você deve decodificar esse texto antes de poder usá-lo AWS CLI para descriptografá-lo. Salva o texto cifrado binário em um arquivo. A parte final do comando \(`> ExampleEncryptedFile`\) salva o texto cifrado binário em um arquivo para facilitar a decodificação. Para ver um exemplo de comando que usa o AWS CLI para descriptografar dados, consulte os exemplos de descriptografia.](#)

Exemplo 2: Usando o AWS CLI para criptografar dados no Windows

Esse exemplo é igual ao exemplo anterior, mas usa a ferramenta `certutil` em vez do `base64`. Esse procedimento requer dois comandos, conforme mostrado no exemplo a seguir.

```
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
```

Exemplo 3: Criptografia com uma chave assimétrica KMS

O `encrypt` comando a seguir mostra como criptografar texto sem formatação com uma chave assimétrica. KMS O parâmetro `--encryption-algorithm` é obrigatório. Como em todos os `encrypt` CLI comandos, o `plaintext` parâmetro deve ser codificado em `base64` ou você deve usar o `fileb://` prefixo, que instrui o AWS CLI a ler dados binários do arquivo.

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [Criptografar](#) na Referência de AWS CLI Comandos.

generate-data-key-pair-without-plaintext

O código de exemplo a seguir mostra como usar `generate-data-key-pair-without-plaintext`.

AWS CLI

Para gerar um par de chaves ECC NIST de dados assimétricos P384

O `generate-data-key-pair-without-plaintext` exemplo a seguir solicita um par de chaves ECC NIST P384 para uso fora do. AWS

O comando retorna uma chave pública em texto simples e uma cópia da chave privada criptografada sob a chave especificada KMS. Ele não retorna uma chave privada de texto simples. Você pode armazenar com segurança a chave privada criptografada com os dados criptografados e ligar AWS KMS para descriptografar a chave privada quando precisar usá-la.

Para solicitar um par de chaves de ECC NIST dados assimétricos P384, use o `key-pair-spec` parâmetro com um valor de `ECC_NIST_P384`

A KMS chave especificada deve ser uma KMS chave de criptografia simétrica, ou seja, uma KMS chave com um KeySpec valor de `SYMMETRIC_DEFAULT`.

NOTE: os valores na saída deste exemplo são truncados para exibição.

```
aws kms generate-data-key-pair-without-plaintext \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-pair-spec ECC_NIST_P384
```

Saída:

```
{  
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkk6Fbh0tQkM1QJJH3PdtHvS/y  
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",  
  "PublicKey":  
  "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrivSn7+Ld1JE1oUoQV5HpEuHAVbd0yND  
+NmYDH/mL10SIEuLrzdZ5hrMH4pk83r401...",
```

```
"KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyPairSpec": "ECC_NIST_P384"  
}
```

Os `PublicKey` e `PrivateKeyCiphertextBlob` são retornados no formato codificado em base64.

Para obter mais informações, consulte [Pares de chaves de dados](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [GenerateDataKeyPairWithoutPlaintext](#) na Referência de AWS CLI Comandos.

generate-data-key-pair

O código de exemplo a seguir mostra como usar `generate-data-key-pair`.

AWS CLI

Para gerar um par de chaves de dados RSA assimétricos de 2048 bits

O `generate-data-key-pair` exemplo a seguir solicita um par de chaves de RSA dados assimétricos de 2048 bits para uso fora do. AWS O comando retorna uma chave pública em texto simples e uma chave privada em texto sem formatação para uso e exclusão imediatos, além de uma cópia da chave privada criptografada sob a chave especificada. KMS Você pode armazenar com segurança a chave privada criptografada com os dados criptografados.

Para solicitar um par de chaves de RSA dados assimétricos de 2048 bits, use `key-pair-spec` o parâmetro com um valor de. `RSA_2048`

A KMS chave especificada deve ser uma KMS chave de criptografia simétrica, ou seja, uma KMS chave com um `KeySpec` valor de `SYMMETRIC_DEFAULT`.

NOTE: os valores na saída deste exemplo são truncados para exibição.

```
aws kms generate-data-key-pair \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-pair-spec RSA_2048
```

Saída:

```
{
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkkK6Fbh0tQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSiB3DQEHBqCCB1...",
  "PrivateKeyPlaintext": "MIIG/
QIBADANBgkqhkiG9w0BAQEFAASCBUcwggbjAgEAAoIBgQDcDd4YzI
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P42ZgMf+YvU5IgS4ut...",
  "PublicKey":
  "MIIB0jANBgkqhkiG9w0BAQEFAAO0CAY8AMIIBigKCAYEA3A3eGMyPrivSn7+Ld1JE1oUoQV5HpEuHAVbd0yND
+NmYDH/mL10SIEuLrcdZ5hrMH4pk83r401...",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "KeyPairSpec": "RSA_2048"
}
```

Os `PublicKeyPrivateKeyPlaintext`, e `PrivateKeyCiphertextBlob` são retornados no formato codificado em base64.

Para obter mais informações, consulte [Pares de chaves de dados](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [GenerateDataKeyPair](#) na Referência de AWS CLI Comandos.

generate-data-key-without-plaintext

O código de exemplo a seguir mostra como usar `generate-data-key-without-plaintext`.

AWS CLI

Como gerar uma chave de dados simétrica de 256 bits sem uma chave de texto simples

O exemplo de `generate-data-key-without-plaintext` a seguir solicita uma cópia criptografada de uma chave de dados simétrica de 256 bits para uso externo à AWS. Você pode ligar AWS KMS para descriptografar a chave de dados quando estiver pronto para usá-la.

Use o parâmetro `key-spec` com um valor de `AES_256` para solicitar uma chave de dados de 256 bits. Use o parâmetro `key-spec` com um valor de `AES_128` para solicitar uma chave de dados de 128 bits. Para todos os outros comprimentos de chave de dados, use o parâmetro `number-of-bytes`.

A KMS chave especificada deve ser uma chave de criptografia KMS simétrica, ou seja, uma KMS chave com um valor de especificação de chave de `SYMMETRIC _ . DEFAULT`

```
aws kms generate-data-key-without-plaintext \
  --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
  --key-spec AES_256
```

Saída:

```
{
  "CiphertextBlob":
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIHvcNAQcGoG8wbQIBADBoBgkqhki
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

A `CiphertextBlob` (chave de dados criptografada) é retornada em um formato codificado em base64.

Para obter mais informações, consulte [Data keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [GenerateDataKeyWithoutPlaintext](#) na Referência de AWS CLI Comandos.

generate-data-key

O código de exemplo a seguir mostra como usar `generate-data-key`.

AWS CLI

Exemplo 1: como gerar uma chave de dados simétrica de 256 bits

O `generate-data-key` exemplo a seguir solicita uma chave de dados simétrica de 256 bits para uso fora do. AWS O comando retorna uma chave de dados em texto simples para uso e exclusão imediatos e uma cópia dessa chave de dados criptografada sob a chave especificada. KMS A chave de dados criptografada pode ser armazenada com segurança junto com os dados criptografados.

Use o parâmetro `key-spec` com um valor de `AES_256` para solicitar uma chave de dados de 256 bits. Use o parâmetro `key-spec` com um valor de `AES_128` para solicitar uma chave de dados de 128 bits. Para todos os outros comprimentos de chave de dados, use o parâmetro `number-of-bytes`.

A KMS chave especificada deve ser uma chave de criptografia KMS simétrica, ou seja, uma KMS chave com um valor de especificação de chave de SYMMETRIC _ . DEFAULT

```
aws kms generate-data-key \
  --key-id alias/ExampleAlias \
  --key-spec AES_256
```

Saída:

```
{
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
  "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "CiphertextBlob":
  "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIHvcNAQcGoG8wbQIBADBoBgkqhki
+YdhV8MrkBQPeac0ReRVNDt9qlEAt+SHgIRF8P0H+7U="
}
```

A Plaintext (chave de dados em texto simples) e a CiphertextBlob (chave de dados criptografada) são retornadas no formato codificado em base64.

Para obter mais informações, consulte Data keys < <https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys> no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: como gerar uma chave de dados simétrica de 512 bits

O exemplo de generate-data-key a seguir solicita uma chave de dados simétrica de 512 bits para criptografia e descryptografia. O comando retorna uma chave de dados em texto simples para uso e exclusão imediatos e uma cópia dessa chave de dados criptografada sob a chave especificada. KMS A chave de dados criptografada pode ser armazenada com segurança junto com os dados criptografados.

Use o parâmetro number-of-bytes para solicitar um comprimento de chave diferente de 128 ou 256 bits. Para solicitar uma chave de dados de 512 bits, o exemplo a seguir usa o parâmetro number-of-bytes com um valor de 64 (bytes).

A KMS chave especificada deve ser uma chave de criptografia KMS simétrica, ou seja, uma KMS chave com um valor de especificação de chave de SYMMETRIC _ . DEFAULT

NOTE: os valores na saída deste exemplo são truncados para exibição.

```
aws kms generate-data-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --number-of-bytes 64
```

Saída:

```
{  
  "CiphertextBlob": "AQIBAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y+hAEnX/  
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSiB3DQEHBqCCA1UwggJRAgEAMIICSgYJKoZ...",  
  "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB  
+G00ZLtf5MSEb4a13R2UKWG0p06njAwy2n72VRm2m7z/  
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Plaintext (chave de dados em texto simples) e CiphertextBlob (chave de dados criptografada) são retornadas no formato codificado em base64.

Para obter mais informações, consulte [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> no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [GenerateDataKey](#) na Referência de AWS CLI Comandos.

generate-random

O código de exemplo a seguir mostra como usar `generate-random`.

AWS CLI

Exemplo 1: Para gerar uma string de bytes aleatória de 256 bits (Linux ou) macOS

O exemplo de `generate-random` a seguir gera uma string de bytes aleatórios de 256 bits (32 bytes) codificada em base64. O exemplo decodifica a string de bytes e a salva no arquivo aleatório.

Ao executar esse comando, use o parâmetro `number-of-bytes` para especificar o tamanho do valor aleatório em bytes.

Você não especifica uma KMS chave ao executar esse comando. A sequência de bytes aleatória não está relacionada a nenhuma KMS chave.

Por padrão, AWS KMS gera o número aleatório. No entanto, se você especificar um armazenamento de chaves personalizadas < <https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html>>, a string de bytes aleatória será gerada no cluster do Cloud associado ao armazenamento de chaves personalizadas. AWS HSM

Este exemplo usa os seguintes parâmetros e valores:

Ele usa o `--number-of-bytes` parâmetro necessário com um valor de 32 para solicitar uma string de 32 bytes (256 bits). Ele usa o `--output` parâmetro com um valor de `text` para direcionar o para retornar a saída como texto, em vez de JSON. Ele usa o AWS CLI para extrair o valor da `Plaintext` propriedade da resposta. Ele canaliza (`|`) a saída do comando `--query` parameter para o `base64` utilitário, que decodifica a saída extraída. Ele usa o operador de redirecionamento (`>`) para salvar a decodificação string de bytes para o arquivo. Ele usa o `ExampleRandom` operador de redirecionamento (`>`) para salvar o texto cifrado binário em um arquivo.

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext | base64 --decode > ExampleRandom
```

Este comando não produz saída.

Para obter mais informações, consulte [GenerateRandom](#) na API Referência do Serviço de Gerenciamento de AWS Chaves.

Exemplo 2: como gerar um número aleatório de 256 bits (prompt de comando do Windows)

O exemplo a seguir usa o comando `generate-random` para gerar uma string aleatória de 256 bits (32 bytes) codificada em base64. O exemplo decodifica a string de bytes e a salva no arquivo aleatório. Esse exemplo é igual ao exemplo anterior, mas usa o utilitário `certutil` no Windows para decodificar em base64 a string de bytes aleatória antes de salvá-la em um arquivo.

Primeiro, gere uma string de bytes aleatória codificada em base64 e salve-a em um arquivo temporário (`ExampleRandom.base64`).

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext > ExampleRandom.base64
```

Como a saída do comando `generate-random` é salva em um arquivo, esse exemplo não produz nenhuma saída.

Use o comando `certutil -decode` para decodificar a string de bytes codificada em base64 no arquivo `ExampleRandom.base64`. Em seguida, ele salva a string de bytes decodificada no arquivo `ExampleRandom`.

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Saída:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Para obter mais informações, consulte [GenerateRandom](#) na API Referência do Serviço de Gerenciamento de AWS Chaves.

- Para API obter detalhes, consulte [GenerateRandom](#) na Referência de AWS CLI Comandos.

get-key-policy

O código de exemplo a seguir mostra como usar `get-key-policy`.

AWS CLI

Para copiar uma política de chaves de uma KMS chave para outra KMS chave

O `get-key-policy` exemplo a seguir obtém a política de chaves de uma KMS chave e a salva em um arquivo de texto. Em seguida, ele substitui a política de uma KMS chave diferente usando o arquivo de texto como entrada da política.

Como o `--policy` parâmetro de `put-key-policy` requer uma string, você deve usar a `--output text` opção de retornar a saída como uma string de texto em vez de JSON.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [PutKeyPolicy](#) na AWS KMS API Referência.

- Para API obter detalhes, consulte [GetKeyPolicy](#) na Referência de AWS CLI Comandos.

get-key-rotation-status

O código de exemplo a seguir mostra como usar `get-key-rotation-status`.

AWS CLI

Para recuperar o status de rotação de uma KMS chave.

O `get-key-rotation-status` exemplo a seguir retorna informações sobre o status de rotação da KMS chave especificada, incluindo se a rotação automática está ativada, o período de rotação e a próxima data de rotação programada. Você pode usar esse comando em KMS chaves gerenciadas pelo cliente e KMS chaves AWS gerenciadas. No entanto, todas as KMS chaves AWS gerenciadas são alternadas automaticamente a cada ano.

```
aws kms get-key-rotation-status \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyRotationEnabled": true,  
  "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",  
  "RotationPeriodInDays": 365  
}
```

Para obter mais informações, consulte [Chaves rotativas](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [GetKeyRotationStatus](#) na Referência de AWS CLI Comandos.

get-parameters-for-import

O código de exemplo a seguir mostra como usar `get-parameters-for-import`.

AWS CLI

Para obter os itens necessários para importar o material chave em uma KMS chave

O `get-parameters-for-import` exemplo a seguir obtém a chave pública e o token de importação necessários para importar o material da chave em uma KMS chave. Ao usar o `import-key-material` comando, certifique-se de usar o token de importação e o material da chave criptografados pela chave pública que foram retornados no mesmo `get-parameters-for-import` comando. Além disso, o algoritmo de encapsulamento especificado nesse comando deve ser usado para criptografar o material da chave com a chave pública.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um ID de chave, mas você pode usar um ID de chave ou chave ARN neste 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
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Baixar a chave pública e importar o token](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [GetParametersForImport](#) na Referência de AWS CLI Comandos.

get-public-key

O código de exemplo a seguir mostra como usar `get-public-key`.

AWS CLI

Exemplo 1: Para baixar a chave pública de uma chave assimétrica KMS

O `get-public-key` exemplo a seguir baixa a chave pública de uma chave assimétrica KMS.

Além de retornar a chave pública, a saída inclui informações das quais você precisa usar a chave pública com segurança fora dela AWS KMS, incluindo o uso da chave e os algoritmos de criptografia compatíveis.

```
aws kms get-public-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "PublicKey": "jANBgqhkhiG9w0BAQEFAAOCAg8AMIICGkCAgEA15epvg1/
QtJhxSi2g9SDEVg8QV/...",
  "CustomerMasterKeySpec": "RSA_4096",
  "KeyUsage": "ENCRYPT_DECRYPT",
  "EncryptionAlgorithms": [
    "RSAES_OAEP_SHA_1",
    "RSAES_OAEP_SHA_256"
  ]
}
```

Para obter mais informações sobre o uso de KMS chaves assimétricas em AWS KMS, consulte [Usando chaves simétricas e assimétricas na Referência do AWS Key Management Service](#). API

Exemplo 2: Para converter uma chave pública em DER formato (Linux e macOS)

O `get-public-key` exemplo a seguir baixa a chave pública de uma KMS chave assimétrica e a salva em um DER arquivo.

Quando você usa o `get-public-key` comando no AWS CLI, ele retorna uma chave pública X.509 DER codificada em -que é codificada em Base64. Este exemplo obtém o valor da `PublicKey` propriedade como texto. Ele decodifica em Base64 `PublicKey` e o salva no arquivo `public_key.der`. O `output` parâmetro retorna a saída como texto, em vez de JSON. O `--query` parâmetro obtém somente a `PublicKey` propriedade, não as propriedades das quais você precisa usar a chave pública com segurança fora dela AWS KMS.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms get-public-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query PublicKey | base64 --decode > public_key.der
```

Este comando não produz saída.

Para obter mais informações sobre o uso de KMS chaves assimétricas em AWS KMS, consulte [Usando chaves simétricas e assimétricas na Referência do AWS Key Management Service](#). API

- Para API obter detalhes, consulte [GetPublicKey](#) na Referência de AWS CLI Comandos.

import-key-material

O código de exemplo a seguir mostra como usar `import-key-material`.

AWS CLI

Para importar material chave em uma KMS chave

O `import-key-material` exemplo a seguir carrega material de chave em uma KMS chave que foi criada sem material de chave. O estado chave da KMS chave deve ser `PendingImport`.

Esse comando usa material de chave que você criptografou com a chave pública que o `get-parameters-for-import` comando retornou. Ele também usa o token de importação do mesmo `get-parameters-for-import` comando.

O `expiration-model` parâmetro indica que o material chave expira automaticamente na data e hora especificadas pelo `valid-to` parâmetro. Quando o material da chave expira, o material da chave é AWS KMS excluído, o estado da chave muda para `Pending import` e a KMS KMS chave fica inutilizável. Para restaurar a KMS chave, você deve reimportar o mesmo material de chave. Para usar um material de chave diferente, você deve criar uma nova KMS chave.

Antes de executar esse comando, substitua o ID da chave de exemplo por um ID de chave válido ou por uma chave ARN da sua AWS conta.

```
aws kms import-key-material \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query PublicKey | base64 --decode > public_key.der
```

```
--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
```

Este comando não produz saída.

Para obter mais informações sobre a importação de material de chaves, consulte [Importando material de chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ImportKeyMaterial](#) na Referência de AWS CLI Comandos.

list-aliases

O código de exemplo a seguir mostra como usar `list-aliases`.

AWS CLI

Exemplo 1: Para listar todos os aliases em uma AWS conta e região

O exemplo a seguir usa o `list-aliases` comando para listar todos os aliases na região padrão da AWS conta. A saída inclui aliases associados às KMS chaves AWS gerenciadas e às KMS chaves gerenciadas pelo cliente.

```
aws kms list-aliases
```

Saída:

```
{  
  "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"
  },
  ...
]
}

```

Exemplo 2: Para listar todos os aliases de uma chave específica KMS

O exemplo a seguir usa o `list-aliases` comando e seu `key-id` parâmetro para listar todos os aliases associados a uma KMS chave específica.

Cada alias está associado a apenas uma KMS chave, mas uma KMS chave pode ter vários aliases. Esse comando é muito útil porque o AWS KMS console lista somente um alias para cada KMS chave. Para encontrar todos os aliases de uma KMS chave, você deve usar o `list-aliases` comando.

Este exemplo usa o ID da KMS chave para o `--key-id` parâmetro, mas você pode usar um ID de chave, chaveARN, nome de alias ou alias ARN nesse comando.

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```

{
  "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"
    }
  ]
}

```



```
]
}
```

Para obter mais informações, consulte [Working with aliases](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ListAliases](#) na Referência de AWS CLI Comandos.

list-grants

O código de exemplo a seguir mostra como usar `list-grants`.

AWS CLI

Para ver as concessões em uma AWS KMS chave

O `list-grants` exemplo a seguir exibe todas as concessões na KMS chave AWS gerenciada especificada para o Amazon DynamoDB em sua conta. Essa concessão permite que o DynamoDB use KMS a chave em seu nome para criptografar uma tabela do DynamoDB antes de gravá-la no disco. Você pode usar um comando como esse para ver as concessões nas KMS chaves AWS gerenciadas e nas KMS chaves gerenciadas pelo cliente na AWS conta e na região.

Esse comando usa o `key-id` parâmetro com um ID de chave para identificar a KMS chave. Você pode usar um ID de chave ou chave ARN para identificar a KMS chave. Para obter o ID da chave ou a chave ARN de uma KMS chave AWS gerenciada, use o `list-aliases` comando `list-keys` ou.

```
aws kms list-grants \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

O resultado mostra que a concessão dá ao Amazon DynamoDB permissão para usar KMS a chave para operações criptográficas, além de permitir que ele visualize detalhes sobre a chave `DescribeKey ()` e KMS retire as concessões `()`. `RetireGrant` A restrição `EncryptionContextSubset` limita a permissão às solicitações que incluem os pares de contexto de criptografia especificados. Como resultado, as permissões na concessão são efetivas somente na conta especificada e na tabela do DynamoDB.

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Concessões AWS KMS no](#) Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ListGrants](#) na Referência de AWS CLI Comandos.

list-key-policies

O código de exemplo a seguir mostra como usar `list-key-policies`.

AWS CLI

Para obter os nomes das principais políticas de uma KMS chave

O exemplo de `list-key-policies` a seguir obtém os nomes das políticas de chave de uma chave gerenciada pelo cliente na conta e região de exemplo. Você pode usar esse comando

para encontrar os nomes das políticas de chaves para chaves AWS gerenciadas e chaves gerenciadas pelo cliente.

Como o único nome de política de chave válido é `default`, esse comando não é muito útil.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou chave ARN neste comando.

```
aws kms list-key-policies \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{  
  "PolicyNames": [  
    "default"  
  ]  
}
```

Para obter mais informações sobre políticas AWS KMS principais, consulte [Usando políticas de chaves AWS KMS no](#) Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ListKeyPolicies](#) na Referência de AWS CLI Comandos.

list-key-rotations

O código de exemplo a seguir mostra como usar `list-key-rotations`.

AWS CLI

Para recuperar informações sobre todas as rotações de materiais principais concluídas

O `list-key-rotations` exemplo a seguir lista informações sobre todas as rotações de material-chave concluídas para a KMS chave especificada.

```
aws kms list-key-rotations \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{
```

```
"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
}
```

Para obter mais informações, consulte [Chaves rotativas](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ListKeyRotations](#) na Referência de AWS CLI Comandos.

list-keys

O código de exemplo a seguir mostra como usar `list-keys`.

AWS CLI

Para obter as KMS chaves em uma conta e região

O `list-keys` exemplo a seguir obtém as KMS chaves em uma conta e região. Esse comando retorna as chaves AWS gerenciadas e as chaves gerenciadas pelo cliente.

```
aws kms list-keys
```

Saída:

```
{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Viewing Keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ListKeys](#) na Referência de AWS CLI Comandos.

list-resource-tags

O código de exemplo a seguir mostra como usar `list-resource-tags`.

AWS CLI

Para obter as etiquetas em uma KMS chave

O `list-resource-tags` exemplo a seguir obtém as tags de uma KMS chave. Para adicionar ou substituir tags de recursos nas KMS teclas, use o `tag-resource` comando. A saída mostra que essa KMS chave tem duas tags de recursos, cada uma com uma chave e um valor.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou chave ARN neste comando.

```
aws kms list-resource-tags \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{
  "Tags": [
    {
      "TagKey": "Dept",
      "TagValue": "IT"
    }
  ]
}
```

```
  },  
  {  
    "TagKey": "Purpose",  
    "TagValue": "Test"  
  }  
],  
"Truncated": false  
}
```

Para obter mais informações sobre o uso de tags em AWS KMS, consulte Como [marcar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ListResourceTags](#) na Referência de AWS CLI Comandos.

list-retirable-grants

O código de exemplo a seguir mostra como usar `list-retirable-grants`.

AWS CLI

Para ver os subsídios que um diretor pode se aposentar

O `list-retirable-grants` exemplo a seguir mostra todas as concessões que o `ExampleAdmin` usuário pode retirar com as KMS chaves em uma AWS conta e região. Você pode usar um comando como este para ver as concessões que qualquer diretor de conta pode retirar com KMS chaves na AWS conta e na região.

O valor do `retiring-principal` parâmetro necessário deve ser o Amazon Resource Name (ARN) de uma conta, usuário ou função.

Você não pode especificar um serviço para o valor de `retiring-principal` neste comando, mesmo que um serviço possa ser o principal que está se aposentando. Para encontrar as concessões nas quais um determinado serviço é o diretor que está se aposentando, use o `list-grants` comando.

O resultado mostra que o `ExampleAdmin` usuário tem permissão para retirar subsídios em duas KMS chaves diferentes na conta e na região. Além do principal que se aposenta, a conta tem permissão para retirar qualquer subsídio na conta.

```
aws kms list-retirable-grants \  
  --retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

Saída:

```
{
  "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":
"8c94d1f12f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
      "Name": "",
      "CreationDate": "2021-02-02T19:49:49.638000+00:00",
      "GranteePrincipal": "arn:aws:iam::111122223333:role/ExampleRole",
      "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
      "IssuingAccount": "arn:aws:iam::111122223333:root",
      "Operations": [
        "Decrypt"
      ],
      "Constraints": {
        "EncryptionContextSubset": {
          "Department": "IT"
        }
      }
    }
  ],
  "Truncated": false
}
```

```
}
```

Para obter mais informações, consulte [Concessões AWS KMS no Guia do desenvolvedor do AWS Key Management Service](#).

- Para API obter detalhes, consulte [ListRetirableGrants](#) na Referência de AWS CLI Comandos.

put-key-policy

O código de exemplo a seguir mostra como usar `put-key-policy`.

AWS CLI

Para alterar a política de chaves para uma KMS chave

O exemplo de `put-key-policy` a seguir altera a política de chave de uma chave gerenciada pelo cliente.

Para começar, crie uma política de chaves e salve-a em um JSON arquivo local. Neste exemplo, o arquivo é `key_policy.json`. Você também pode especificar a política de chave como o valor da string do parâmetro `policy`.

A primeira declaração nessa política de chaves dá permissão à AWS conta para usar IAM políticas para controlar o acesso à KMS chave. A segunda instrução dá permissão ao `test-user` usuário para executar os `list-keys` comandos `describe-key` e na KMS tecla.

Conteúdo de `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" : "*"
    }
]
}

```

Para identificar a KMS chave, este exemplo usa o ID da chave, mas você também pode usar uma chaveARN. O comando usa o parâmetro `policy` para especificar a política de chave. Para indicar que a política está em um arquivo, ela usa o prefixo `file://` necessário. Esse prefixo é necessário para identificar arquivos em todos os sistemas operacionais compatíveis. Por fim, o comando usa o parâmetro `policy-name` com um valor `default`. Se nenhum nome de política for especificado, o valor padrão será `default`. O único valor válido é `default`.

```

aws kms put-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --policy file://key_policy.json

```

Esse comando não produz nenhuma saída. Use o comando `get-key-policy` para verificar se o comando está em vigor. O comando de exemplo a seguir obtém a política de chaves para a mesma KMS chave. O parâmetro `output` com o valor `text` retorna um formato de texto fácil de ler.

```

aws kms get-key-policy \
  --policy-name default \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --output text

```

Saída:

```

{
  "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" : "*"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Changing a Key Policy](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [PutKeyPolicy](#) na Referência de AWS CLI Comandos.

re-encrypt

O código de exemplo a seguir mostra como usar `re-encrypt`.

AWS CLI

Exemplo 1: Para recriptografar uma mensagem criptografada com uma KMS chave simétrica diferente (Linux e macOS).

O exemplo de `re-encrypt` comando a seguir demonstra a forma recomendada de recriptografar dados com o AWS CLI

Forneça o texto cifrado em um arquivo. No valor do `--ciphertext-blob` parâmetro, use o `fileb://` prefixo, que instrui o CLI a ler os dados de um arquivo binário. Se o arquivo

não estiver no diretório atual, digite o caminho completo para o arquivo. Para obter mais informações sobre a leitura de valores de AWS CLI parâmetros de um arquivo, consulte Carregando AWS CLI parâmetros de um arquivo < <https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-parameters-file.html> > no Guia do usuário da interface de linha de AWS comando e práticas recomendadas para parâmetros de arquivos locais < <https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/> > no blog da ferramenta de linha de comando. Especifique a chave de origem, que descriptografa o texto cifrado. O parâmetro não é necessário ao descriptografar com AWS simetria KMS --source-key-id KMSchaves de criptografia. AWS KMS pode obter a KMS chave que foi usada para criptografar os dados dos metadados no blob de texto cifrado. Mas é sempre uma boa prática especificar a KMS chave que você está usando. Essa prática garante que você use a KMS chave desejada e evita que você decodifique inadvertidamente um texto cifrado usando uma KMS chave na qual você não confia. Especifique a chave de destino KMS, que criptografa novamente os dados. O parâmetro é sempre obrigatório. --destination-key-id Este exemplo usa uma chave ARN, mas você pode usar qualquer identificador de chave válido. Solicite a saída de texto sem formatação como um valor de texto. O --query parâmetro diz CLI ao para obter somente o valor do campo da saída. Plaintext O parâmetro --output retorna a saída como texto. Decodifique o texto simples em Base64 e salve-o em um arquivo. O seguinte exemplo canaliza (|) o valor do parâmetro Plaintext para o utilitário Base64, que o decodifica. Em seguida, ele redireciona (>) a saída decodificada para o arquivo ExamplePlaintext.

Antes de executar esse comando, substitua a chave IDs de exemplo por identificadores de chave válidos da sua AWS conta.

```
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
```

Este comando não produz saída. A saída do comando re-encrypt é decodificada em base64 e salva em um arquivo.

Para obter mais informações, consulte ReEncrypt < https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html na Referência do AWS Key Management Service. API

Exemplo 2: Para recriptografar uma mensagem criptografada com uma KMS chave simétrica diferente (prompt de comando do Windows).

O exemplo de comando `re-encrypt` a seguir é o mesmo que o anterior, mas ele usa o utilitário `certutil` para decodificar em Base64 os dados em texto simples. Esse procedimento requer dois comandos, conforme mostrado nos exemplos a seguir.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
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
```

Em seguida, use o utilitário `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Saída:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Para obter mais informações, consulte `ReEncrypt` < https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html na Referência do AWS Key Management Service. API

- Para API obter detalhes, consulte [ReEncrypt](#) na Referência de AWS CLI Comandos.

retire-grant

O código de exemplo a seguir mostra como usar `retire-grant`.

AWS CLI

Como descontinuar uma concessão em uma chave mestra do cliente

O `retire-grant` exemplo a seguir exclui uma concessão de uma KMS chave.

O comando de exemplo a seguir especifica os parâmetros `grant-id` e `key-id`. O valor do `key-id` parâmetro deve ser a chave ARN da KMS chave.

```
aws kms retire-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Use o comando `list-grants` para confirmar a descontinuação da concessão.

Para obter mais informações, consulte [Retiring and revoking grants](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [RetireGrant](#) na Referência de AWS CLI Comandos.

revoke-grant

O código de exemplo a seguir mostra como usar `revoke-grant`.

AWS CLI

Como revogar uma concessão em uma chave mestra do cliente

O `revoke-grant` exemplo a seguir exclui uma concessão de uma KMS chave. O comando de exemplo a seguir especifica os parâmetros `grant-id` e `key-id`. O valor do `key-id` parâmetro pode ser o ID da chave ou a chave ARN da KMS chave.

```
aws kms revoke-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Use o comando `list-grants` para confirmar que a concessão foi revogada.

Para obter mais informações, consulte [Retiring and revoking grants](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [RevokeGrant](#) na Referência de AWS CLI Comandos.

rotate-key-on-demand

O código de exemplo a seguir mostra como usar `rotate-key-on-demand`.

AWS CLI

Para realizar a rotação sob demanda de uma chave KMS

O `rotate-key-on-demand` exemplo a seguir inicia imediatamente a rotação do material da chave para a KMS chave especificada.

```
aws kms rotate-key-on-demand \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Saída:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Para obter mais informações, consulte [Como realizar a rotação de chaves sob demanda](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [RotateKeyOnDemand](#) na Referência de AWS CLI Comandos.

schedule-key-deletion

O código de exemplo a seguir mostra como usar `schedule-key-deletion`.

AWS CLI

Para agendar a exclusão de uma KMS chave gerenciada pelo cliente.

O `schedule-key-deletion` exemplo a seguir agenda a KMS chave gerenciada pelo cliente especificada para ser excluída em 15 dias.

O `--key-id` parâmetro identifica a KMS chave. Este exemplo usa um ARN valor de chave, mas você pode usar o ID da chave ou o ARN da KMS chave. O `--pending-window-in-days` parâmetro especifica a duração do período de espera de 7 a 30 dias. Por padrão, o período de espera é de 30 dias. Este exemplo especifica um valor de 15, que indica AWS a exclusão permanente da KMS chave 15 dias após a conclusão do comando.

```
aws kms schedule-key-deletion \  
  --key-id arn:aws:kms:us-east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

```
--key-id arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
--pending-window-in-days 15
```

A resposta inclui a chaveARN, o estado da chave, o período de espera (PendingWindowInDays) e a data de exclusão no horário Unix. Para ver a data de exclusão no horário local, use o AWS KMS console. KMSchaves no estado PendingDeletion chave não podem ser usadas em operações criptográficas.

```
{  
  "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  
}
```

Para obter mais informações, consulte [Deleting keys](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [ScheduleKeyDeletion](#) na Referência de AWS CLI Comandos.

sign

O código de exemplo a seguir mostra como usar sign.

AWS CLI

Exemplo 1: Para gerar uma assinatura digital para uma mensagem

O sign exemplo a seguir gera uma assinatura criptográfica para uma mensagem curta. A saída do comando inclui um Signature campo codificado em base 64 que você pode verificar usando o comando. `verify`

Você deve especificar uma mensagem para assinar e um algoritmo de assinatura compatível com sua KMS chave assimétrica. Para obter os algoritmos de assinatura da sua KMS chave, use o `describe-key` comando.

Na AWS CLI versão 2.0, o valor do message parâmetro deve ser codificado em Base64. Ou você pode salvar a mensagem em um arquivo e usar o `fileb://` prefixo, que instrui o usuário AWS CLI a ler os dados binários do arquivo.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta. O ID da chave deve representar uma KMS chave assimétrica com um uso de chave de 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
```

Saída:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Para obter mais informações sobre o uso de KMS chaves assimétricas em AWS KMS, consulte Chaves [assimétricas AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

Exemplo 2: Para salvar uma assinatura digital em um arquivo (Linux emacOs)

O sign exemplo a seguir gera uma assinatura criptográfica para uma mensagem curta armazenada em um arquivo local. O comando também obtém a Signature propriedade da resposta, decodifica em Base64 e a salva no arquivo. ExampleSignature Você pode usar o arquivo de assinatura em um verify comando que verifica a assinatura.

O sign comando requer uma mensagem codificada em Base64 e um algoritmo de assinatura compatível com sua chave assimétrica. KMS Para obter os algoritmos de assinatura compatíveis com sua KMS chave, use o describe-key comando.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta. O ID da chave deve representar uma KMS chave assimétrica com um uso de chave de 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
```

Este comando não produz saída. Este exemplo extrai a `Signature` propriedade da saída e a salva em um arquivo.

Para obter mais informações sobre o uso de KMS chaves assimétricas em AWS KMS, consulte Chaves [assimétricas AWS KMS no Guia](#) do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte Referência do AWS CLI comando de [login](#).

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a uma KMS chave

O `tag-resource` exemplo a seguir adiciona `"Purpose": "Test"` e `"Dept": "IT"` marca uma KMS chave gerenciada pelo cliente. Você pode usar tags como essas para rotular KMS chaves e criar categorias de KMS chaves para permissões e auditoria.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou chave ARN neste comando.

```
aws kms tag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

Este comando não produz saída. Para ver as tags em uma AWS KMS KMS chave, use o `list-resource-tags` comando.

Para obter mais informações sobre o uso de tags em AWS KMS, consulte Como [marcar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para excluir uma tag de uma KMS chave

O `untag-resource` exemplo a seguir exclui a tag com a "Purpose" chave de uma KMS chave gerenciada pelo cliente.

Para especificar a KMS chave, use o `key-id` parâmetro. Este exemplo usa um valor de ID de chave, mas você pode usar um ID de chave ou chave ARN neste comando. Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
aws kms untag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tag-key 'Purpose'
```

Este comando não produz saída. Para ver as tags em uma AWS KMS KMS chave, use o `list-resource-tags` comando.

Para obter mais informações sobre o uso de tags em AWS KMS, consulte Como [marcar chaves](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-alias

O código de exemplo a seguir mostra como usar `update-alias`.

AWS CLI

Para associar um alias a uma chave diferente KMS

O `update-alias` exemplo a seguir associa o alias `alias/test-key` a uma chave diferente KMS.

O parâmetro `--alias-name` especifica o alias. O valor do nome do alias deve começar com `alias/`. O `--target-key-id` parâmetro especifica a KMS chave a ser associada ao alias. Você não precisa especificar a KMS chave atual para o alias.

```
aws kms update-alias \  
  --alias-name alias/test-key \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Este comando não produz saída. Use o `list-aliases` comando para localizar o alias.

Para obter mais informações, consulte [Updating aliases](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [UpdateAlias](#) na Referência de AWS CLI Comandos.

update-custom-key-store

O código de exemplo a seguir mostra como usar `update-custom-key-store`.

AWS CLI

Exemplo 1: Para editar o nome amigável de um armazenamento de chaves personalizado

O `update-custom-key-store` exemplo a seguir altera o nome do armazenamento de chaves personalizadas. Este exemplo funciona para um armazenamento de HSM chaves do AWS Cloud ou um armazenamento de chaves externo.

Use o `custom-key-store-id` para identificar o armazenamento de chaves. Use o `new-custom-key-store-name` parâmetro para especificar o novo nome amigável.

Para atualizar o nome amigável de um armazenamento de HSM chaves do AWS Cloud, você deve primeiro desconectar o armazenamento de chaves, por exemplo, usando o `disconnect-custom-key-store` comando. Você pode atualizar o nome amigável de um armazenamento de chaves externo enquanto ele estiver conectado ou desconectado. Para encontrar o estado da conexão do seu armazenamento de chaves personalizadas, use o `describe-custom-key-store` comando.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --new-custom-key-store-name ExampleKeyStore
```

Esse comando não retorna nenhum dado. Para verificar se o comando funcionou, use um `describe-custom-key-stores` comando.

Para obter mais informações sobre a atualização de um armazenamento de HSM chaves do AWS Cloud, consulte [Editando as configurações do armazenamento de HSM chaves do AWS Cloud](#) no Guia do desenvolvedor do AWS Key Management Service.

Para obter mais informações sobre a atualização de um armazenamento de chaves externo, consulte [Edição das propriedades do armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 2: para editar a senha `kmsuser` de um AWS armazenamento de chaves do Cloud HSM

O `update-custom-key-store` exemplo a seguir atualiza o valor da `kmsuser` senha para a senha atual do HSM cluster `kmsuser` in the Cloud associado ao armazenamento de chaves especificado. Esse comando não altera a `kmsuser` senha do cluster. Ele apenas informa AWS KMS a senha atual. Se KMS não tiver a `kmsuser` senha atual, não poderá se conectar ao armazenamento de HSM chaves do AWS Cloud.

NOTE: antes de atualizar um armazenamento de HSM chaves do AWS Cloud, você deve desconectá-lo. Use o comando `disconnect-custom-key-store`. Depois que o comando for concluído, você poderá reconectar o AWS armazenamento de HSM chaves do Cloud. Use o comando `connect-custom-key-store`.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --key-store-password ExamplePassword
```

Esse comando não retorna nenhuma saída. Para verificar se a alteração foi efetiva, use um `describe-custom-key-stores` comando.

Para obter mais informações sobre a atualização de um armazenamento de HSM chaves do AWS Cloud, consulte [Editando as configurações do armazenamento de HSM chaves do AWS Cloud](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 3: Para editar o HSM cluster de AWS nuvem de um armazenamento de HSM chaves do AWS Cloud

O exemplo a seguir altera o HSM cluster do AWS Cloud associado a um armazenamento de HSM chaves do AWS Cloud para um cluster relacionado, como um backup diferente do mesmo cluster.

NOTE: antes de atualizar um armazenamento de HSM chaves do AWS Cloud, você deve desconectá-lo. Use o comando `disconnect-custom-key-store`. Depois que o comando for concluído, você poderá reconectar o AWS armazenamento de HSM chaves do Cloud. Use o comando `connect-custom-key-store`.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

Esse comando não retorna nenhuma saída. Para verificar se a alteração foi efetiva, use um `describe-custom-key-stores` comando.

Para obter mais informações sobre a atualização de um armazenamento de HSM chaves do AWS Cloud, consulte [Editando as configurações do armazenamento de HSM chaves do AWS Cloud](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 4: Para editar a credencial de autenticação de proxy de um armazenamento de chaves externo

O exemplo a seguir atualiza a credencial de autenticação de proxy para seu armazenamento de chaves externo. Você deve especificar o `raw-secret-access-key` e `oaccess-key-id`, mesmo se estiver alterando somente um dos valores. Você pode usar esse recurso para corrigir uma credencial inválida ou para alterar a credencial quando o proxy externo do armazenamento de chaves a gira.

Estabeleça a credencial de autenticação de proxy AWS KMS em seu armazenamento de chaves externo. Em seguida, use esse comando para fornecer a credencial a. AWS KMS AWS KMSusa essa credencial para assinar suas solicitações no proxy externo do armazenamento de chaves.

Você pode atualizar a credencial de autenticação do proxy enquanto o armazenamento de chaves externo estiver conectado ou desconectado. Para encontrar o estado da conexão do seu armazenamento de chaves personalizadas, use o `describe-custom-key-store` comando.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
  RawSecretAccessKey=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Esse comando não retorna nenhuma saída. Para verificar se a alteração foi efetiva, use um `describe-custom-key-stores` comando.

Para obter mais informações sobre a atualização de um armazenamento de chaves externo, consulte [Edição das propriedades do armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

Exemplo 5: Para editar a conectividade proxy de um armazenamento de chaves externo

O exemplo a seguir altera a opção de conectividade proxy do armazenamento de chaves externo de conectividade de endpoint público para conectividade de serviço de VPC endpoint. Além de alterar o `xks-proxy-connectivity` valor, você deve alterar o `xks-proxy-uri-endpoint` valor para refletir o DNS nome privado associado ao serviço de VPC endpoint. Você também deve agregar um `xks-proxy-vpc-endpoint-service-name` valor.

NOTE: antes de atualizar a conectividade proxy de uma loja externa, você deve desconectá-la. Use o comando `disconnect-custom-key-store`. Depois que o comando for concluído, você poderá reconectar o armazenamento de chaves externo usando o `connect-custom-key-store` comando.

```
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"
```

Esse comando não retorna nenhuma saída. Para verificar se a alteração foi efetiva, use um `describe-custom-key-stores` comando.

Para obter mais informações sobre a atualização de um armazenamento de chaves externo, consulte [Edição das propriedades do armazenamento de chaves externo](#) no Guia do desenvolvedor do AWS Key Management Service.

- Para API obter detalhes, consulte [UpdateCustomKeyStore](#) na Referência de AWS CLI Comandos.

update-key-description

O código de exemplo a seguir mostra como usar `update-key-description`.

AWS CLI

Exemplo 1: Para adicionar ou alterar uma descrição em uma KMS chave gerenciada pelo cliente

O `update-key-description` exemplo a seguir adiciona uma descrição a uma KMS chave gerenciada pelo cliente. Você pode usar o mesmo comando para alterar uma descrição existente.

O `--key-id` parâmetro identifica a KMS chave no comando. Este exemplo usa um ARN valor de chave, mas você pode usar o ID da chave ou a chave ARN da KMS chave. O `--description` parâmetro especifica a nova descrição. O valor desse parâmetro substitui a descrição atual da KMS chave, se houver.

```
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"
```

Este comando não produz saída. Para ver a descrição de uma KMS chave, use o `describe-key` comando.

Para obter mais informações, consulte [UpdateKeyDescription](#) na API Referência do Serviço de Gerenciamento de AWS Chaves.

Exemplo 2: Para excluir a descrição de uma KMS chave gerenciada pelo cliente

O `update-key-description` exemplo a seguir exclui a descrição de uma KMS chave gerenciada pelo cliente.

O `--key-id` parâmetro identifica a KMS chave no comando. Este exemplo usa um valor de ID de chave, mas você pode usar o ID da chave ou a chave ARN da KMS chave. O `--description` parâmetro com um valor de cadeia de caracteres vazio ("") exclui a descrição existente.

```
aws kms update-key-description \  
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --description ''
```

Este comando não produz saída. Para ver a descrição de uma KMS chave, use o comando `describe-key`.

Para obter mais informações, consulte [UpdateKeyDescription](#) na API Referência do Serviço de Gerenciamento de AWS Chaves.

- Para API obter detalhes, consulte [UpdateKeyDescription](#) na Referência de AWS CLI Comandos.

verify

O código de exemplo a seguir mostra como usar `verify`.

AWS CLI

Para verificar uma assinatura digital

O `verify` exemplo a seguir verifica uma assinatura criptográfica para uma mensagem curta codificada em Base64. O ID da chave, a mensagem, o tipo de mensagem e o algoritmo de assinatura devem ser os mesmos usados para assinar a mensagem. A assinatura que você especifica não pode ser codificada em base64. Para obter ajuda na decodificação da assinatura que o `sign` comando retorna, consulte os exemplos de `sign` comandos.

A saída do comando inclui um `SignatureValid` campo booleano que indica que a assinatura foi verificada. Se a validação da assinatura falhar, o `verify` comando também falhará.

Antes de executar esse comando, substitua o ID de chave de exemplo por um ID de chave válido da sua AWS conta.

```
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
```

Saída:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "SignatureValid": true,  
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"  
}
```

Para obter mais informações sobre o uso de KMS chaves assimétricas em AWS KMS, consulte [Usando chaves assimétricas no AWS Key Management Service Developer Guide](#).

- Para API obter detalhes, consulte [Verificar](#) na Referência de AWS CLI Comandos.

Exemplos de Lake Formation usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Lake Formation.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-lf-tags-to-resource

O código de exemplo a seguir mostra como usar `add-lf-tags-to-resource`.

AWS CLI

Para anexar uma ou mais etiquetas LF a um recurso existente

O `add-lf-tags-to-resource` exemplo a seguir anexa a tag LF fornecida ao recurso da tabela.

```
aws lakeformation add-lf-tags-to-resource \  
--cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  }  
}
```

```
  },
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
      "analyst"
    ]
  }]
}
```

Saída:

```
{
  "Failures": []
}
```

Para obter mais informações, consulte [Atribuição de tags LF aos recursos do catálogo de dados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [AddLfTagsToResource](#) na Referência de AWS CLI Comandos.

batch-grant-permissions

O código de exemplo a seguir mostra como usar batch-grant-permissions.

AWS CLI

Para conceder permissões em massa sobre recursos aos diretores

O batch-grant-permissions exemplo a seguir concede acesso em massa a recursos específicos aos diretores.

```
aws lakeformation batch-grant-permissions \
  --cli-input-json file://input.json
```

Conteúdo de 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": []
  }
]
}

```

Saída:

```

{
  "Failures": []
}

```

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no](#) Guia do Desenvolvedor do AWS Lake Formation.

- Para API obter detalhes, consulte [BatchGrantPermissions](#) na Referência de AWS CLI Comandos.

batch-revoke-permissions

O código de exemplo a seguir mostra como usar batch-revoke-permissions.

AWS CLI

Para revogar em massa as permissões de recursos dos diretores

O batch-revoke-permissions exemplo a seguir revoga em massa o acesso dos diretores a recursos específicos.

```
aws lakeformation batch-revoke-permissions \  
  --cli-input-json file://input.json
```

Conteúdo de 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"
      ]
    }
  ]
}
```

Saída:

```
{
  "Failures": []
}
```

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no Guia do Desenvolvedor do AWS Lake Formation](#).

- Para API obter detalhes, consulte [BatchRevokePermissions](#) na Referência de AWS CLI Comandos.

cancel-transaction

O código de exemplo a seguir mostra como usar `cancel-transaction`.

AWS CLI

Para cancelar uma transação

O `cancel-transaction` exemplo a seguir cancela a transação.

```
aws lakeformation cancel-transaction \  
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Este comando não produz saída.

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [CancelTransaction](#) na Referência de AWS CLI Comandos.

commit-transaction

O código de exemplo a seguir mostra como usar `commit-transaction`.

AWS CLI

Para confirmar a transação

O `commit-transaction` exemplo a seguir confirma a transação.

```
aws lakeformation commit-transaction \  
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Saída:

```
{  
  "TransactionStatus": "committed"  
}
```

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [CommitTransaction](#) na Referência de AWS CLI Comandos.

create-data-cells-filter

O código de exemplo a seguir mostra como usar `create-data-cells-filter`.

AWS CLI

Exemplo 1: Para criar um filtro de células de dados

O `create-data-cells-filter` exemplo a seguir cria um filtro de célula de dados para permitir que se conceda acesso a determinadas colunas com base na condição da linha.

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

Conteúdo de `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"  
  }  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Filtragem de dados e segurança em nível de célula no Lake Formation no Lake Formation Developer AWS Guide](#).

Exemplo 2: Para criar um filtro de coluna

O `create-data-cells-filter` exemplo a seguir cria um filtro de dados para permitir que alguém conceda acesso a determinadas colunas.

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

Conteúdo de `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"
  }
}

```

Este comando não produz saída.

Para obter mais informações, consulte [Filtragem de dados e segurança em nível de célula no Lake Formation no Lake Formation Developer AWS Guide](#).

Exemplo 3: Para criar um filtro de dados com colunas de exclusão

O `create-data-cells-filter` exemplo a seguir cria um filtro de dados para permitir que se conceda acesso a todas, exceto às colunas mencionadas.

```

aws lakeformation create-data-cells-filter \
  --cli-input-json file://input.json

```

Conteúdo de `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"
  }
}

```

Este comando não produz saída.

Para obter mais informações, consulte [Filtragem de dados e segurança em nível de célula no Lake Formation no Lake Formation Developer AWS Guide](#).

- Para API obter detalhes, consulte [CreateDataCellsFilter](#) na Referência de AWS CLI Comandos.

create-lf-tag

O código de exemplo a seguir mostra como usar `create-lf-tag`.

AWS CLI

Para criar uma etiqueta LF

O `create-lf-tag` exemplo a seguir cria uma tag LF com o nome e os valores especificados.

```
aws lakeformation create-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values ['developer','analyst','campaign']
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags LF para controle de acesso a metadados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [CreateLfTag](#) na Referência de AWS CLI Comandos.

delete-data-cells-filter

O código de exemplo a seguir mostra como usar `delete-data-cells-filter`.

AWS CLI

Para excluir o filtro de células de dados

O `delete-data-cells-filter` exemplo a seguir exclui um determinado filtro de célula de dados.

```
aws lakeformation delete-data-cells-filter \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "TableCatalogId": "123456789111",
  "DatabaseName": "tpc",
  "TableName": "dl_tpc_promotion",
  "Name": "developer_promotion"
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Filtragem de dados e segurança em nível de célula no Lake Formation no Lake Formation Developer AWS Guide](#).

- Para API obter detalhes, consulte [DeleteDataCellsFilter](#) na Referência de AWS CLI Comandos.

delete-lf-tag

O código de exemplo a seguir mostra como usar delete-lf-tag.

AWS CLI

Para excluir a definição da tag LF

O delete-lf-tag exemplo a seguir exclui a definição da tag LF.

```
aws lakeformation delete-lf-tag \
  --catalog-id '123456789111' \
  --tag-key 'usergroup'
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags LF para controle de acesso a metadados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [DeleteLfTag](#) na Referência de AWS CLI Comandos.

delete-objects-on-cancel

O código de exemplo a seguir mostra como usar delete-objects-on-cancel.

AWS CLI

Para excluir o objeto quando a transação é cancelada

O `delete-objects-on-cancel` exemplo a seguir exclui o objeto s3 listado quando a transação é cancelada.

```
aws lakeformation delete-objects-on-cancel \  
--cli-input-json file://input.json
```

Conteúdo de `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"  
  }]  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [DeleteObjectsOnCancel](#) na Referência de AWS CLI Comandos.

deregister-resource

O código de exemplo a seguir mostra como usar `deregister-resource`.

AWS CLI

Para cancelar o registro do armazenamento em data lake

O `deregister-resource` exemplo a seguir cancela o registro do recurso conforme gerenciado pelo Lake Formation.

```
aws lakeformation deregister-resource \  
--cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma localização do Amazon S3 ao seu data lake no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [DeregisterResource](#) na Referência de AWS CLI Comandos.

describe-transaction

O código de exemplo a seguir mostra como usar `describe-transaction`.

AWS CLI

Para recuperar os detalhes de uma transação

O `describe-transaction` exemplo a seguir retorna os detalhes de uma única transação.

```
aws lakeformation describe-transaction \
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Saída:

```
{
  "TransactionDescription": {
    "TransactionId": "12345972ca8347b89825e33c5774aec4",
    "TransactionStatus": "committed",
    "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
    "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
  }
}
```

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [DescribeTransaction](#) na Referência de AWS CLI Comandos.

extend-transaction

O código de exemplo a seguir mostra como usar `extend-transaction`.

AWS CLI

Para estender uma transação

O `extend-transaction` exemplo a seguir estende a transação.

```
aws lakeformation extend-transaction \  
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Este comando não produz saída.

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [ExtendTransaction](#) na Referência de AWS CLI Comandos.

get-data-lake-settings

O código de exemplo a seguir mostra como usar `get-data-lake-settings`.

AWS CLI

Para recuperar as configurações do data AWS lake gerenciado pelo Lake Formation

O `get-data-lake-settings` exemplo a seguir recupera a lista de administradores de data lake e outras configurações de data lake.

```
aws lakeformation get-data-lake-settings \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "CatalogId": "123456789111"  
}
```

Saída:

```
{
  "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"
    ]
  }
}
```

Para obter mais informações, consulte [Alterando as configurações de segurança padrão do seu data lake no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [GetDataLakeSettings](#) na Referência de AWS CLI Comandos.

get-effective-permissions-for-path

O código de exemplo a seguir mostra como usar `get-effective-permissions-for-path`.

AWS CLI

Para recuperar permissões em recursos localizados em um caminho específico

O `get-effective-permissions-for-path` exemplo a seguir retorna as permissões do Lake Formation para uma tabela ou recurso de banco de dados especificado localizado em um caminho no Amazon S3.

```
aws lakeformation get-effective-permissions-for-path \  
--cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"  
}
```

Saída:

```
{  
  "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":
  "E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWLP
}

```

Para obter mais informações, consulte [Gerenciamento de permissões do Lake Formation](#) no Guia do desenvolvedor do AWS Lake Formation.

- Para API obter detalhes, consulte [GetEffectivePermissionsForPathna](#) Referência de AWS CLI Comandos.

get-lf-tag

O código de exemplo a seguir mostra como usar `get-lf-tag`.

AWS CLI

Para recuperar a definição da tag LF

O `get-lf-tag` exemplo a seguir recupera a definição da tag LF.

```

aws lakeformation get-lf-tag \
  --catalog-id '123456789111' \
  --tag-key 'usergroup'

```

Saída:

```

{
  "CatalogId": "123456789111",

```

```
"TagKey": "usergroup",
"TagValues": [
  "analyst",
  "campaign",
  "developer"
]
}
```

Para obter mais informações, consulte [Gerenciamento de tags LF para controle de acesso a metadados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [GetLfTag](#) na Referência de AWS CLI Comandos.

get-query-state

O código de exemplo a seguir mostra como usar `get-query-state`.

AWS CLI

Para recuperar o estado de uma consulta enviada

O `get-query-state` exemplo a seguir retorna o estado de uma consulta enviada anteriormente.

```
aws lakeformation get-query-state \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Saída:

```
{
  "State": "FINISHED"
}
```

Para obter mais informações, consulte [Operações de dados transacionais](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [GetQueryState](#) na Referência de AWS CLI Comandos.

get-query-statistics

O código de exemplo a seguir mostra como usar `get-query-statistics`.

AWS CLI

Para recuperar estatísticas de consulta

O `get-query-statistics` exemplo a seguir recupera estatísticas sobre o planejamento e a execução de uma consulta.

```
aws lakeformation get-query-statistics \  
--query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Operações de dados transacionais](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [GetQueryStatistics](#) na Referência de AWS CLI Comandos.

get-resource-lf-tags

O código de exemplo a seguir mostra como usar `get-resource-lf-tags`.

AWS CLI

Para listar etiquetas LF

O `list-lf-tags` exemplo a seguir retorna uma lista de tags LF que o solicitante tem permissão para visualizar.

```
aws lakeformation list-lf-tags \  
  --cli-input-json file://input.json
```

Conteúdo de input.json:

```
{  
  "CatalogId": "123456789111",  
  "ResourceShareType": "ALL",  
  "MaxResults": 2  
}
```

Saída:

```
{  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "category",  
    "TagValues": [  
      "private",  
      "public"  
    ]  
  },  
  {  
    "CatalogId": "123456789111",  
    "TagKey": "group",  
    "TagValues": [  
      "analyst",  
      "campaign",  
      "developer"  
    ]  
  }  
],  
  "NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWVvbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"  
}
```

Para obter mais informações, consulte [Gerenciamento de tags LF para controle de acesso a metadados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [GetResourceLfTags](#) na Referência de AWS CLI Comandos.

get-table-objects

O código de exemplo a seguir mostra como usar get-table-objects.

AWS CLI

Para listar objetos de uma tabela controlada

O `get-table-objects` exemplo a seguir retorna o conjunto de objetos do Amazon S3 que compõem a tabela controlada especificada.

```
aws lakeformation get-table-objects \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_household_demographics_gov",  
  "QueryAsOfTime": "2022-08-10T15:00:00"  
}
```

Saída:

```
{  
  "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  
    }]  
  }]  
}
```

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [GetTableObjects](#) na Referência de AWS CLI Comandos.

get-work-unit-results

O código de exemplo a seguir mostra como usar `get-work-unit-results`.

AWS CLI

Para recuperar unidades de trabalho de uma determinada consulta

O `get-work-unit-results` exemplo a seguir retorna as unidades de trabalho resultantes da consulta.

```
aws lakeformation get-work-units \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \  
  --work-unit-id '0' \  
  --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

Saída:

```
outfile with Blob content.
```

Para obter mais informações, consulte [Operações de dados transacionais](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [GetWorkUnitResults](#) na Referência de AWS CLI Comandos.

`get-work-units`

O código de exemplo a seguir mostra como usar `get-work-units`.

AWS CLI

Para recuperar unidades de trabalho

O `get-work-units` exemplo a seguir recupera as unidades de trabalho geradas pela `StartQueryPlanning` operação.

```
aws lakeformation get-work-units \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Saída:

```
{  
  "WorkUnitRanges": [{  
    "WorkUnitIdMax": 0,  
    "WorkUnitIdMin": 0,
```

```

    "WorkUnitToken":
      "1234eMAk4kL04umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
+f88jzGrYq22gE6jkQ1p0B
+0et2eqNumFudAAAAfjB8BkgqhkIG9w0BBwagbzBtAgEAMGgGCSqGSIB3DQEHATAeBg1ghkgBZQMEAS4wEQQMCOEWRda
wAAAAEAAAAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmB1
+VSojiG0tdsUk7vcjYXUUboYm3dvdqRqX2s4gROM0n
+Ij8R0/8jYmnHkpvYAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytm0hozxD0Ei/
F2ZoXz6wLGDfGUo/2WxCkY0hTyNaw6TM
+7drTM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzB1AjBH8okRtYP8R7AY2W1s/
hqFBhG0V4l42AC0LxsuZbMQrE2SzWZUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCnk1ExMR
+btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
    },
    "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
  }

```

Para obter mais informações, consulte [Operações de dados transacionais](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [GetWorkUnits](#) na Referência de AWS CLI Comandos.

grant-permissions

O código de exemplo a seguir mostra como usar `grant-permissions`.

AWS CLI

Exemplo 1: Para conceder permissões ao diretor sobre recursos usando tags LF

O `grant-permissions` exemplo a seguir concede ALL permissões ao principal no recurso de banco de dados que corresponde à política de tag LF.

```

aws lakeformation grant-permissions \
  --cli-input-json file://input.json

```

Conteúdo de `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"
    ]
  }
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no](#) Guia do Desenvolvedor do AWS Lake Formation.

Exemplo 2: Para conceder permissões em nível de coluna ao diretor

O `grant-permissions` exemplo a seguir concede permissão para selecionar uma coluna específica para o diretor.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Conteúdo de `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": []
}

```

Este comando não produz saída.

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no](#) Guia do Desenvolvedor do AWS Lake Formation.

Exemplo 3: Para conceder permissões de tabela ao diretor

O `grant-permissions` exemplo a seguir concede permissão de seleção em todas as tabelas de um determinado banco de dados ao principal.

```

aws lakeformation grant-permissions \
  --cli-input-json file://input.json

```

Conteúdo de `input.json`:

```

{
  "CatalogId": "123456789111",
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
  },
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "TableWildcard": {}
    }
  },
  "Permissions": [
    "SELECT"
  ],
  "PermissionsWithGrantOption": []
}

```

```
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no Guia do Desenvolvedor do AWS Lake Formation](#).

Exemplo 4: Para conceder permissões em etiquetas LF ao diretor

O `grant-permissions` exemplo a seguir concede permissão de associado em etiquetas LF ao diretor.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Conteúdo de `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": []  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no Guia do Desenvolvedor do AWS Lake Formation](#).

Exemplo 5: Para conceder permissões sobre localizações de dados ao diretor

O `grant-permissions` exemplo a seguir concede permissão sobre a localização dos dados ao diretor.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Conteúdo de `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": []  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no](#) Guia do Desenvolvedor do AWS Lake Formation.

- Para API obter detalhes, consulte [GrantPermissions](#) na Referência de AWS CLI Comandos.

list-data-cells-filter

O código de exemplo a seguir mostra como usar `list-data-cells-filter`.

AWS CLI

Para listar filtros de células de dados

O `list-data-cells-filter` exemplo a seguir lista o filtro de células de dados para determinada tabela.

```
aws lakeformation list-data-cells-filter \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "MaxResults": 2,  
  "Table": {  
    "CatalogId": "123456789111",  
    "DatabaseName": "tpc",  
    "Name": "dl_tpc_promotion"  
  }  
}
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Filtragem de dados e segurança em nível de célula no Lake Formation no Lake Formation Developer AWS Guide](#).

- Para API obter detalhes, consulte [ListDataCellsFilter](#) na Referência de AWS CLI Comandos.

list-permissions

O código de exemplo a seguir mostra como usar `list-permissions`.

AWS CLI

Exemplo 1: Para recuperar a lista das principais permissões do recurso

O `list-permissions` exemplo a seguir retorna uma lista das principais permissões nos recursos do banco de dados.

```
aws lakeformation list-permissions \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "CatalogId": "123456789111",
  "ResourceType": "DATABASE",
  "MaxResults": 2
}
```

Saída:

```
{
  "PrincipalResourcePermissions": [{
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
    },
    "Resource": {
      "Database": {
        "CatalogId": "123456789111",
        "Name": "tpc"
      }
    },
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  }],
  "NextToken":
  "E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dU1qcDdJbk5sWTI5dVpITW1P
}
```

Para obter mais informações, consulte [Gerenciamento de permissões do Lake Formation](#) no Guia do desenvolvedor do AWS Lake Formation.

Exemplo 2: Para recuperar a lista das principais permissões na tabela com filtros de dados

O `list-permissions` exemplo a seguir lista as permissões na tabela com filtros de dados relacionados concedidos ao diretor.

```
aws lakeformation list-permissions \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "CatalogId": "123456789111",
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
```

```

        "Name": "dl_tpc_customer"
      }
    },
    "IncludeRelated": "TRUE",
    "MaxResults": 10
  }

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [Gerenciamento de permissões do Lake Formation](#) no Guia do desenvolvedor do AWS Lake Formation.

Exemplo 3: Para recuperar a lista das principais permissões nas tags LF

O `list-permissions` exemplo a seguir lista as permissões nas etiquetas LF concedidas ao diretor.

```
aws lakeformation list-permissions \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "LFTag": {  
      "CatalogId": "123456789111",  
      "TagKey": "category",  
      "TagValues": [  
        "private"  
      ]  
    }  
  },  
  "MaxResults": 10  
}
```

Saída:

```
{  
  "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="
}

```

Para obter mais informações, consulte [Gerenciamento de permissões do Lake Formation](#) no Guia do desenvolvedor do AWS Lake Formation.

- Para API obter detalhes, consulte [ListPermissions](#) na Referência de AWS CLI Comandos.

list-resources

O código de exemplo a seguir mostra como usar `list-resources`.

AWS CLI

Para listar os recursos gerenciados pela Lake Formation

O `list-resources` exemplo a seguir lista os recursos que correspondem à condição gerenciada pelo Lake Formation.

```
aws lakeformation list-resources \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "FilterConditionList": [{  
    "Field": "ROLE_ARN",  
    "ComparisonOperator": "CONTAINS",  
    "StringValueList": [  
      "123456789111"  
    ]  
  }],  
  "MaxResults": 10  
}
```

Saída:

```
{  
  "ResourceInfoList": [{  
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111",  
    "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole",  
    "LastModified": "2022-07-21T02:12:46.669000+00:00"  
  },  
  {  
    "ResourceArn": "arn:aws:s3:::lf-emr-test-123456789111",  
    "RoleArn": "arn:aws:iam::123456789111:role/EMRIFS3Role",  
    "LastModified": "2022-07-29T16:22:03.211000+00:00"  
  }  
]  
}
```

Para obter mais informações, consulte [Gerenciamento de permissões do Lake Formation](#) no Guia do desenvolvedor do AWS Lake Formation.

- Para API obter detalhes, consulte [ListResources](#) na Referência de AWS CLI Comandos.

list-transactions

O código de exemplo a seguir mostra como usar `list-transactions`.

AWS CLI

Para listar todos os detalhes das transações

O `list-transactions` exemplo a seguir retorna metadados sobre transações e seu status.

```
aws lakeformation list-transactions \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "CatalogId": "123456789111",  
  "StatusFilter": "ALL",  
  "MaxResults": 3  
}
```

Saída:

```
{  
  "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=="
}
```

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [ListTransactions](#) na Referência de AWS CLI Comandos.

put-data-lake-settings

O código de exemplo a seguir mostra como usar `put-data-lake-settings`.

AWS CLI

Para definir as configurações AWS do data lake gerenciado pelo Lake Formation

O `put-data-lake-settings` exemplo a seguir define a lista de administradores de data lake e outras configurações de data lake.

```
aws lakeformation put-data-lake-settings \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "DataLakeSettings": {
    "DataLakeAdmins": [{
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
    }
  ],
  "CreateDatabaseDefaultPermissions": [],
  "CreateTableDefaultPermissions": [],
  "TrustedResourceOwners": [],
  "AllowExternalDataFiltering": true,
  "ExternalDataFilteringAllowList": [{
    "DataLakePrincipalIdentifier": "123456789111"
  }],
  "AuthorizedSessionTagValueList": ["Amazon EMR"]
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Alterando as configurações de segurança padrão do seu data lake no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [PutDataLakeSettings](#) na Referência de AWS CLI Comandos.

register-resource

O código de exemplo a seguir mostra como usar `register-resource`.

AWS CLI

Exemplo 1: Para registrar o armazenamento do data lake usando o Service Linked Role

O `register-resource` exemplo a seguir registra o recurso como gerenciado pelo Lake Formation usando a função vinculada ao serviço.

```
aws lakeformation register-resource \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",  
  "UseServiceLinkedRole": true  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma localização do Amazon S3 ao seu data lake no AWS Lake Formation Developer Guide](#).

Exemplo 2: Para registrar o armazenamento do data lake usando uma função personalizada

O `register-resource` exemplo a seguir registra o recurso como gerenciado pelo Lake Formation usando uma função personalizada.

```
aws lakeformation register-resource \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
  "UseServiceLinkedRole": false,
  "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar uma localização do Amazon S3 ao seu data lake no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [RegisterResource](#) na Referência de AWS CLI Comandos.

remove-**lf-tags-from-resource**

O código de exemplo a seguir mostra como usar `remove-lf-tags-from-resource`.

AWS CLI

Para remover o LF-tag de um recurso

O `remove-lf-tags-from-resource` exemplo a seguir remove a associação da tag LF com o recurso de tabela.

```
aws lakeformation remove-lf-tags-from-resource \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "CatalogId": "123456789111",
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_promotion"
    }
  },
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
```



```
        "developer"
      ]
    }]
  }
```

Saída:

```
{
  "Failures": []
}
```

Para obter mais informações, consulte [Atribuição de tags LF aos recursos do catálogo de dados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [RemoveLfTagsFromResource](#) na Referência de AWS CLI Comandos.

revoke-permissions

O código de exemplo a seguir mostra como usar `revoke-permissions`.

AWS CLI

Para revogar permissões sobre recursos do diretor

O `revoke-permissions` exemplo a seguir revoga o acesso principal à tabela específica de um determinado banco de dados.

```
aws lakeformation revoke-permissions \
  --cli-input-json file://input.json
```

Conteúdo de `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": []
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Conceder e revogar permissões nos recursos do Catálogo de Dados no](#) Guia do Desenvolvedor do AWS Lake Formation.

- Para API obter detalhes, consulte [RevokePermissions](#) na Referência de AWS CLI Comandos.

search-databases-by-lf-tags

O código de exemplo a seguir mostra como usar `search-databases-by-lf-tags`.

AWS CLI

Para pesquisar recursos de banco de dados por LFTags

O `search-databases-by-lf-tags` exemplo a seguir pesquisa na LFTag expressão correspondente de recursos de banco de dados.

```
aws lakeformation search-databases-by-lf-tags \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "MaxResults": 1,
  "CatalogId": "123456789111",
  "Expression": [{
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
}
```

```
}
```

Saída:

```
{
  "DatabaseList": [{
    "Database": {
      "CatalogId": "123456789111",
      "Name": "tpc"
    },
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }
  ]
}]
}
```

Para obter mais informações, consulte [Visualização dos recursos aos quais uma tag LF é atribuída](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [SearchDatabasesByLfTags](#) na Referência de AWS CLI Comandos.

search-tables-by-lf-tags

O código de exemplo a seguir mostra como usar `search-tables-by-lf-tags`.

AWS CLI

Para pesquisar os recursos da tabela por LFTags

O `search-tables-by-lf-tags` exemplo a seguir pesquisa na LFTag expressão correspondente de recursos de tabela.

```
aws lakeformation search-tables-by-lf-tags \
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{
  "MaxResults": 2,
  "CatalogId": "123456789111",
  "Expression": [{
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
}
```

Saída:

```
{
  "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"
        ]
    }]
}
]
}]
}
```

Para obter mais informações, consulte [Visualização dos recursos aos quais uma tag LF é atribuída](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [SearchTablesByLfTags](#) na Referência de AWS CLI Comandos.

start-query-planning

O código de exemplo a seguir mostra como usar `start-query-planning`.

AWS CLI

Para processar a declaração de consulta

O `start-query-planning` exemplo a seguir envia uma solicitação para processar uma declaração de consulta.

```
aws lakeformation start-query-planning \  
  --cli-input-json file://input.json
```

Conteúdo de `input.json`:

```
{  
  "QueryPlanningContext": {  
    "CatalogId": "012345678901",  
    "DatabaseName": "tpc"  
  },  
  "QueryString": "select * from dl_tpc_household_demographics_gov where  
hd_income_band_sk=9"  
}
```

Saída:

```
{  
  "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"  
}
```

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [StartQueryPlanning](#) na Referência de AWS CLI Comandos.

start-transaction

O código de exemplo a seguir mostra como usar `start-transaction`.

AWS CLI

Para iniciar uma nova transação

O `start-transaction` exemplo a seguir inicia uma nova transação e retorna seu ID de transação.

```
aws lakeformation start-transaction \  
  --transaction-type = 'READ_AND_WRITE'
```

Saída:

```
{  
  "TransactionId": "b014d972ca8347b89825e33c5774aec4"  
}
```

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [StartTransaction](#) na Referência de AWS CLI Comandos.

update-lf-tag

O código de exemplo a seguir mostra como usar `update-lf-tag`.

AWS CLI

Para atualizar a definição da tag LF

O `update-lf-tag` exemplo a seguir atualiza a definição da tag LF.

```
aws lakeformation update-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values-to-add '['admin']'
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags LF para controle de acesso a metadados](#) no AWS Lake Formation Developer Guide.

- Para API obter detalhes, consulte [UpdateLFTag](#) na Referência de AWS CLI Comandos.

update-table-objects

O código de exemplo a seguir mostra como usar `update-table-objects`.

AWS CLI

Para modificar objetos de uma tabela controlada

O `update-table-objects` exemplo a seguir adiciona objetos S3 fornecidos à tabela controlada especificada.

```
aws lakeformation update-table-objects \  
  --cli-input-json file://input.json
```

Conteúdo de `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  
    }  
  }  
}]  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Lendo e gravando no data lake dentro das transações no AWS Lake Formation Developer Guide](#).

- Para API obter detalhes, consulte [UpdateTableObjects](#) na Referência de AWS CLI Comandos.

Exemplos de Lambda usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Lambda.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-layer-version-permission

O código de exemplo a seguir mostra como usar `add-layer-version-permission`.

AWS CLI

Para adicionar permissões a uma versão de camada

O `add-layer-version-permission` exemplo a seguir concede permissão para a conta especificada usar a versão 1 da camada `my-layer`.

```
aws lambda add-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --action Lambda:GetLayerVersion \  
  --principal 123456789012 \  
  --version-number 1
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [AddLayerVersionPermission](#) na Referência de AWS CLI Comandos.

add-permission

O código de exemplo a seguir mostra como usar `add-permission`.

AWS CLI

Para adicionar permissões a uma função Lambda existente

O `add-permission` exemplo a seguir concede ao SNS serviço da Amazon permissão para invocar uma função chamada `my-function`.

```

aws lambda add-permission \
  --function-name my-function \
  --action lambda:InvokeFunction \
  --statement-id sns \
  --principal sns.amazonaws.com

```

Saída:

```

{
  "Statement":
  {
    "Sid": "sns",
    "Effect": "Allow",
    "Principal": {
      "Service": "sns.amazonaws.com"
    }
  }
}

```

```
    },
    "Action": "lambda:InvokeFunction",
    "Resource": "arn:aws:lambda:us-east-2:123456789012:function:my-function"
  }
}
```

Para obter mais informações, consulte [Usando políticas baseadas em recursos para o Lambda AWS no Guia do desenvolvedor](#) do AWS Lambda.

- Para API obter detalhes, consulte [AddPermission](#) na Referência de AWS CLI Comandos.

create-alias

O código de exemplo a seguir mostra como usar `create-alias`.

AWS CLI

Para criar um alias para uma função do Lambda

O seguinte exemplo de `create-alias` cria um alias chamado LIVE que aponta para a versão 1 da função `my-function` do Lambda.

```
aws lambda create-alias \
  --function-name my-function \
  --description "alias for live version of function" \
  --function-version 1 \
  --name LIVE
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [CreateAlias](#) na Referência de AWS CLI Comandos.

create-event-source-mapping

O código de exemplo a seguir mostra como usar `create-event-source-mapping`.

AWS CLI

Para criar um mapeamento entre uma fonte de eventos e uma função AWS Lambda

O `create-event-source-mapping` exemplo a seguir cria um mapeamento entre uma SQS fila e a função `my-function` Lambda.

```
aws lambda create-event-source-mapping \  
  --function-name my-function \  
  --batch-size 5 \  
  --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [AWS Lambda Event Source Mapping](#) no AWS Lambda Developer Guide.

- Para API obter detalhes, consulte [CreateEventSourceMapping](#) na Referência de AWS CLI Comandos.

create-function

O código de exemplo a seguir mostra como usar `create-function`.

AWS CLI

Criar uma função do Lambda

O exemplo `create-function` a seguir cria uma função do Lambda denominada `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
```

Conteúdo de `my-function.zip`:

This file is a deployment package that contains your function code and any dependencies.

Saída:

```
{  
  "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": ""  
}
```

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [CreateFunction](#) na Referência de AWS CLI Comandos.

delete-alias

O código de exemplo a seguir mostra como usar `delete-alias`.

AWS CLI

Para excluir um alias de uma função do Lambda

O seguinte exemplo de `delete-alias` exclui o alias chamado LIVE da função `my-function` do Lambda.

```
aws lambda delete-alias \  
  --function-name my-function \  
  --name LIVE
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [DeleteAlias](#) na Referência de AWS CLI Comandos.

delete-event-source-mapping

O código de exemplo a seguir mostra como usar `delete-event-source-mapping`.

AWS CLI

Para excluir o mapeamento entre uma fonte de eventos e uma função AWS Lambda

O `delete-event-source-mapping` exemplo a seguir exclui o mapeamento entre uma SQS fila e a função `Lambdamy-function`.

```
aws lambda delete-event-source-mapping \  
  --uuid a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [AWS Lambda Event Source Mapping](#) no AWS Lambda Developer Guide.

- Para API obter detalhes, consulte [DeleteEventSourceMapping](#) na Referência de AWS CLI Comandos.

delete-function-concurrency

O código de exemplo a seguir mostra como usar `delete-function-concurrency`.

AWS CLI

Para remover o limite reservado de execução simultânea de uma função

O seguinte exemplo de `delete-function-concurrency` exclui o limite reservado de execução simultânea da função `my-function`.

```
aws lambda delete-function-concurrency \  
  --function-name my-function
```

Este comando não produz saída.

Para obter mais informações, consulte [Como reservar simultaneidade para uma função do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [DeleteFunctionConcurrency](#) na Referência de AWS CLI Comandos.

delete-function-event-invoke-config

O código de exemplo a seguir mostra como usar `delete-function-event-invoke-config`.

AWS CLI

Para excluir uma configuração de invocação assíncrona

O `delete-function-event-invoke-config` exemplo a seguir exclui a configuração de invocação assíncrona para o GREEN alias da função especificada.

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- Para API obter detalhes, consulte [DeleteFunctionEventInvokeConfig](#) na Referência de AWS CLI Comandos.

delete-function

O código de exemplo a seguir mostra como usar `delete-function`.

AWS CLI

Exemplo 1: excluir uma função do Lambda pelo nome da função

O exemplo de `delete-function` a seguir exclui a função do Lambda denominada `my-function` ao especificar o nome da função.

```
aws lambda delete-function \  
  --function-name my-function
```

Este comando não produz saída.

Exemplo 2: Para excluir uma função Lambda por função ARN

O `delete-function` exemplo a seguir exclui a função Lambda `my-function` nomeada especificando a função. ARN

```
aws lambda delete-function \  
  --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Este comando não produz saída.

Exemplo 3: Para excluir uma função Lambda por meio de uma função parcial ARN

O `delete-function` exemplo a seguir exclui a função Lambda `my-function` nomeada especificando a parcial da função. ARN

```
aws lambda delete-function \  
  --function-name 123456789012:function:my-function
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [DeleteFunction](#) na Referência de AWS CLI Comandos.

delete-layer-version

O código de exemplo a seguir mostra como usar `delete-layer-version`.

AWS CLI

Para excluir uma versão de uma camada Lambda

O `delete-layer-version` exemplo a seguir exclui a versão 2 da camada chamada `my-layer`.

```
aws lambda delete-layer-version \  
  --layer-name my-layer \  
  --version-number 2
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [DeleteLayerVersion](#) na Referência de AWS CLI Comandos.

delete-provisioned-concurrency-config

O código de exemplo a seguir mostra como usar `delete-provisioned-concurrency-config`.

AWS CLI

Para excluir uma configuração de simultaneidade provisionada

O seguinte exemplo de `delete-provisioned-concurrency-config` exclui a configuração de simultaneidade provisionada para o alias GREEN da função especificada.

```
aws lambda delete-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier GREEN
```

- Para API obter detalhes, consulte [DeleteProvisionedConcurrencyConfig](#)na Referência de AWS CLI Comandos.

get-account-settings

O código de exemplo a seguir mostra como usar `get-account-settings`.

AWS CLI

Para recuperar detalhes sobre sua conta em uma região AWS

O seguinte exemplo de `get-account-settings` mostra os limites do Lambda e as informações de uso da sua conta.

```
aws lambda get-account-settings
```

Saída:

```
{
  "AccountLimit": {
    "CodeSizeUnzipped": 262144000,
    "UnreservedConcurrentExecutions": 1000,
    "ConcurrentExecutions": 1000,
    "CodeSizeZipped": 52428800,
    "TotalCodeSize": 80530636800
  },
  "AccountUsage": {
    "FunctionCount": 4,
    "TotalCodeSize": 9426
  }
}
```

Para obter mais informações, consulte [Limites do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [GetAccountSettings](#)na Referência de AWS CLI Comandos.

get-alias

O código de exemplo a seguir mostra como usar `get-alias`.

AWS CLI

Para recuperar detalhes sobre um alias de função

O seguinte exemplo de `get-alias` mostra os detalhes do alias chamado LIVE na função `my-function` do Lambda.

```
aws lambda get-alias \  
  --function-name my-function \  
  --name LIVE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [GetAlias](#) na Referência de AWS CLI Comandos.

`get-event-source-mapping`

O código de exemplo a seguir mostra como usar `get-event-source-mapping`.

AWS CLI

Para recuperar detalhes sobre um mapeamento de origem de eventos

O `get-event-source-mapping` exemplo a seguir exibe os detalhes do mapeamento entre uma SQS fila e a função `my-function` Lambda.

```
aws lambda get-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [AWS Lambda Event Source Mapping](#) no AWS Lambda Developer Guide.

- Para API obter detalhes, consulte [GetEventSourceMapping](#) na Referência de AWS CLI Comandos.

get-function-concurrency

O código de exemplo a seguir mostra como usar `get-function-concurrency`.

AWS CLI

Para visualizar a configuração de simultaneidade reservada para uma função

O seguinte exemplo de `get-function-concurrency` recupera a configuração de simultaneidade reservada para a função especificada.

```
aws lambda get-function-concurrency \
  --function-name my-function
```

Saída:

```
{
  "ReservedConcurrentExecutions": 250
}
```

- Para API obter detalhes, consulte [GetFunctionConcurrency](#) na Referência de AWS CLI Comandos.

get-function-configuration

O código de exemplo a seguir mostra como usar `get-function-configuration`.

AWS CLI

Para recuperar as configurações específicas da versão de uma função do Lambda

O seguinte exemplo de `get-function-configuration` mostra as configurações da versão 2 da função `my-function`.

```
aws lambda get-function-configuration \  
  --function-name my-function:2
```

Saída:

```
{  
  "FunctionName": "my-function",  
  "LastModified": "2019-09-26T20:28:40.438+0000",  
  "RevisionId": "e52502d4-9320-4688-9cd6-152a6ab7490d",  
  "MemorySize": 256,  
  "Version": "2",  
  "Role": "arn:aws:iam::123456789012:role/service-role/my-function-role-uy3l9qq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJm1KidWoaCgk=",  
  "Description": "",  
  "VpcConfig": {  
    "SubnetIds": [],  
    "VpcId": "",  
    "SecurityGroupIds": []  
  },  
  "CodeSize": 304,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",  
  "Handler": "index.handler"  
}
```

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [GetFunctionConfiguration](#) na Referência de AWS CLI Comandos.

get-function-event-invoke-config

O código de exemplo a seguir mostra como usar `get-function-event-invoke-config`.

AWS CLI

Para visualizar uma configuração de invocação assíncrona

O `get-function-event-invoke-config` exemplo a seguir recupera a configuração de invocação assíncrona para o BLUE alias da função especificada.

```
aws lambda get-function-event-invoke-config \  
  --function-name my-function:BLUE
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetFunctionEventInvokeConfig](#) na Referência de AWS CLI Comandos.

get-function

O código de exemplo a seguir mostra como usar `get-function`.

AWS CLI

Recuperar informações sobre uma função

O exemplo `get-function` a seguir mostra informações sobre a função `my-function`.

```
aws lambda get-function \  
  --function-name my-function
```

Saída:

```
{  
  "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/0J1FrzJmlKidWaaCgk=",  
    "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": ""  
  }  
}
```

```
}  
}
```

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [GetFunction](#) na Referência de AWS CLI Comandos.

get-layer-version-by-arn

O código de exemplo a seguir mostra como usar `get-layer-version-by-arn`.

AWS CLI

Para recuperar informações sobre uma versão da camada Lambda

O `get-layer-version-by-arn` exemplo a seguir exibe informações sobre a versão da camada com o Amazon Resource Name (ARN) especificado.

```
aws lambda get-layer-version-by-arn \  
  --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-  
  SciPy1x:2"
```

Saída:

```
{  
  "LayerVersionArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-  
  Python311-SciPy1x:2",  
  "Description": "AWS Lambda SciPy layer for Python 3.11 (scipy-1.1.0,  
  numpy-1.15.4) https://github.com/scipy/scipy/releases/tag/v1.1.0 https://  
  github.com/numpy/numpy/releases/tag/v1.15.4",  
  "CreateDate": "2023-10-12T10:09:38.398+0000",  
  "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-  
  SciPy1x",  
  "Content": {  
    "CodeSize": 41784542,  
    "CodeSha256": "GGmv8ocUw4cly0T8HL0Vx/f5V4RmSCGNjDIslY4VskM=",  
    "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/  
  snapshots/123456789012/..."  
  },  
  "Version": 2,  
  "CompatibleRuntimes": [  
    "python3.11"  ]  
}
```

```
    "python3.11"  
  ],  
  "LicenseInfo": "SciPy: https://github.com/scipy/scipy/blob/main/LICENSE.txt,  
  NumPy: https://github.com/numpy/numpy/blob/main/LICENSE.txt"  
}
```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [GetLayerVersionByArna](#) Referência de AWS CLI Comandos.

get-layer-version-policy

O código de exemplo a seguir mostra como usar `get-layer-version-policy`.

AWS CLI

Para recuperar a política de permissões para uma versão da camada Lambda

O `get-layer-version-policy` exemplo a seguir exibe informações de política sobre a versão 1 para a camada chamada `my-layer`.

```
aws lambda get-layer-version-policy \  
  --layer-name my-layer \  
  --version-number 1
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [GetLayerVersionPolicy](#) na Referência de AWS CLI Comandos.

get-layer-version

O código de exemplo a seguir mostra como usar `get-layer-version`.

AWS CLI

Para recuperar informações sobre uma versão da camada Lambda

O `get-layer-version` exemplo a seguir exibe informações para a versão 1 da camada chamada `my-layer`.

```
aws lambda get-layer-version \  
  --layer-name my-layer \  
  --version-number 1
```

Saída:

```
{  
  "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"  
  ]  
}
```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [GetLayerVersion](#) na Referência de AWS CLI Comandos.

get-policy

O código de exemplo a seguir mostra como usar `get-policy`.

AWS CLI

Para recuperar a IAM política baseada em recursos para uma função, versão ou alias

O seguinte exemplo de `get-policy` mostra informações de política sobre a função `my-function` do Lambda.

```
aws lambda get-policy \  
  --function-name my-function
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Usando políticas baseadas em recursos para o Lambda AWS no Guia do desenvolvedor](#) do AWS Lambda.

- Para API obter detalhes, consulte [GetPolicy](#) na Referência de AWS CLI Comandos.

get-provisioned-concurrency-config

O código de exemplo a seguir mostra como usar `get-provisioned-concurrency-config`.

AWS CLI

Para visualizar uma configuração de simultaneidade provisionada

O seguinte exemplo de `get-provisioned-concurrency-config` mostra detalhes da configuração de simultaneidade provisionada para o alias BLUE da função especificada.

```
aws lambda get-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier BLUE
```

Saída:

```
{  
  "RequestedProvisionedConcurrentExecutions": 100,  
  "AvailableProvisionedConcurrentExecutions": 100,  
  "AllocatedProvisionedConcurrentExecutions": 100,  
  "Status": "READY",  
  "LastModified": "2019-12-31T20:28:49+0000"  
}
```

- Para API obter detalhes, consulte [GetProvisionedConcurrencyConfig](#) na Referência de AWS CLI Comandos.

invoke

O código de exemplo a seguir mostra como usar `invoke`.

AWS CLI

Exemplo 1: invocar uma função do Lambda de forma síncrona

O exemplo de `invoke` a seguir invoca a função `my-function` de forma síncrona. A `cli-binary-format` opção é necessária se você estiver usando a AWS CLI versão 2. Para obter mais informações, consulte as [opções globais de linha de comando AWS CLI suportadas](#) no Guia do usuário da interface de linha de comando de AWS.

```
aws lambda invoke \  
  --function-name my-function \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

Saída:

```
{  
  "ExecutedVersion": "$LATEST",  
  "StatusCode": 200  
}
```

Para obter mais informações, consulte [Invocação síncrona](#) no Guia do desenvolvedor do AWS Lambda.

Exemplo 2: invocar uma função do Lambda de forma assíncrona

O exemplo `invoke` a seguir invoca a função `my-function` de forma assíncrona. A `cli-binary-format` opção é necessária se você estiver usando a AWS CLI versão 2. Para obter mais informações, consulte as [opções globais de linha de comando AWS CLI suportadas](#) no Guia do usuário da interface de linha de comando de AWS.

```
aws lambda invoke \  
  --function-name my-function \  
  --invocation-type Event \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

Saída:

```
{  
  "StatusCode": 202  
}
```

Para obter mais informações, consulte [Invocação assíncrona](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [Invoke](#) in AWS CLI Command Reference.

list-aliases

O código de exemplo a seguir mostra como usar `list-aliases`.

AWS CLI

Para recuperar a lista de aliases para uma função Lambda

O `list-aliases` exemplo a seguir exibe uma lista dos aliases da função `my-function` Lambda.

```
aws lambda list-aliases \  
  --function-name my-function
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [ListAliases](#) na Referência de AWS CLI Comandos.

list-event-source-mappings

O código de exemplo a seguir mostra como usar `list-event-source-mappings`.

AWS CLI

Para listar os mapeamentos de origem de eventos para uma função

O `list-event-source-mappings` exemplo a seguir exibe uma lista dos mapeamentos de origem de eventos para a função `Lambdamy-function`.

```
aws lambda list-event-source-mappings \
  --function-name my-function
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [AWS Lambda Event Source Mapping](#) no AWS Lambda Developer Guide.

- Para API obter detalhes, consulte [ListEventSourceMappings](#) na Referência de AWS CLI Comandos.

list-function-event-invoke-configs

O código de exemplo a seguir mostra como usar `list-function-event-invoke-configs`.

AWS CLI

Para ver uma lista de configurações de invocação assíncrona

O `list-function-event-invoke-configs` exemplo a seguir lista as configurações de invocação assíncrona para a função especificada.

```
aws lambda list-function-event-invoke-configs \
  --function-name my-function
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListFunctionEventInvokeConfigs](#) na Referência de AWS CLI Comandos.

list-functions

O código de exemplo a seguir mostra como usar `list-functions`.

AWS CLI

Recuperar uma lista de funções do Lambda

O exemplo de `list-functions` a seguir exibe uma lista de todas as funções do usuário atual.

```
aws lambda list-functions
```

Saída:

```
{
  "Functions": [
    {
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "Version": "$LATEST",
      "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbl/RMr0fT/I=",
      "FunctionName": "helloworld",
      "MemorySize": 128,
      "RevisionId": "1718e831-badf-4253-9518-d0644210af7b",
      "CodeSize": 294,
      "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:helloworld",
      "Handler": "helloworld.handler",
      "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",
      "Timeout": 3,
      "LastModified": "2023-09-23T18:32:33.857+0000",
      "Runtime": "nodejs18.x",
      "Description": ""
    },
    {
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "Version": "$LATEST",
      "CodeSha256": "sU0cJ2/h0ZevwV/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",
    "Handler": "index.handler",
    "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qqq",
    "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": ""
    }
]
}
```

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [ListFunctions](#) na Referência de AWS CLI Comandos.

list-layer-versions

O código de exemplo a seguir mostra como usar `list-layer-versions`.

AWS CLI

Para listar as versões de uma camada AWS Lambda

O `list-layers-versions` exemplo a seguir exibe informações sobre as versões da camada chamada `my-layer`.

```
aws lambda list-layer-versions \
  --layer-name my-layer
```

Saída:

```
{
  "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"
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [ListLayerVersions](#) na Referência de AWS CLI Comandos.

list-layers

O código de exemplo a seguir mostra como usar `list-layers`.

AWS CLI

Para listar as camadas que são compatíveis com o tempo de execução da sua função

O `list-layers` exemplo a seguir exibe informações sobre camadas que são compatíveis com o tempo de execução do Python 3.11.

```
aws lambda list-layers \
  --compatible-runtime python3.11
```

Saída:

```
{
  "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"
        ]
      }
    }
  ]
}
```



```
]
}
```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [ListLayers](#) na Referência de AWS CLI Comandos.

list-provisioned-concurrency-configs

O código de exemplo a seguir mostra como usar `list-provisioned-concurrency-configs`.

AWS CLI

Para obter uma lista de configurações de simultaneidade provisionada

O seguinte exemplo de `list-provisioned-concurrency-configs` lista as configurações de simultaneidade provisionada para a função especificada.

```
aws lambda list-provisioned-concurrency-configs \
  --function-name my-function
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [ListProvisionedConcurrencyConfigs](#) na Referência de AWS CLI Comandos.

list-tags

O código de exemplo a seguir mostra como usar `list-tags`.

AWS CLI

Para recuperar a lista de tags para uma função do Lambda

O seguinte exemplo de `list-tags` mostra as tags anexadas à função `my-function` do Lambda.

```
aws lambda list-tags \
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Saída:

```
{
  "Tags": {
    "Category": "Web Tools",
    "Department": "Sales"
  }
}
```

Para obter mais informações, consulte [Como marcar funções do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [ListTags](#) na Referência de AWS CLI Comandos.

list-versions-by-function

O código de exemplo a seguir mostra como usar `list-versions-by-function`.

AWS CLI

Para recuperar uma lista de versões de uma função

O seguinte exemplo de `list-versions-by-function` mostra a lista de versões da função `my-function` do Lambda.

```
aws lambda list-versions-by-function \  
--function-name my-function
```

Saída:

```
{  
  "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": "5tT2qqzYUHoqwR616pZ2dpkn/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-uy3l9qqq",
    "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-uy3l9qqq",
    "Timeout": 3,
    "LastModified": "2019-10-01T16:47:28.490+0000",
    "Runtime": "nodejs10.x",
    "Description": "newer version"
}
]
```

```
}
```

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [ListVersionsByFunction](#) na Referência de AWS CLI Comandos.

publish-layer-version

O código de exemplo a seguir mostra como usar `publish-layer-version`.

AWS CLI

Para criar uma versão da camada Lambda

O `publish-layer-version` exemplo a seguir cria uma nova versão da camada da biblioteca Python. O comando recupera o conteúdo da camada de um arquivo chamado `layer.zip` no bucket S3 especificado.

```
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
```

Saída:

```
{  
  "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"
]
}
```

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [PublishLayerVersion](#) na Referência de AWS CLI Comandos.

publish-version

O código de exemplo a seguir mostra como usar `publish-version`.

AWS CLI

Para publicar uma nova versão de uma função

O seguinte exemplo de `publish-version` publica uma nova versão da função `my-function` do Lambda.

```
aws lambda publish-version \
  --function-name my-function
```

Saída:

```
{
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "CodeSha256": "dBG9m8SGdm1Ejw/JYX1hhvCrAv5TxvXsbL/RM1r0fT/I=",
  "FunctionName": "my-function",
  "CodeSize": 294,
  "RevisionId": "f31d3d39-cc63-4520-97d4-43cd44c94c20",
  "MemorySize": 128,
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:3",
  "Version": "2",
  "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-zgur6bf4",
  "Timeout": 3,
  "LastModified": "2019-09-23T18:32:33.857+0000",
```

```
"Handler": "my-function.handler",
"Runtime": "nodejs10.x",
"Description": ""
}
```

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [PublishVersion](#) na Referência de AWS CLI Comandos.

put-function-concurrency

O código de exemplo a seguir mostra como usar `put-function-concurrency`.

AWS CLI

Para configurar um limite de simultaneidade reservado para uma função

O seguinte exemplo de `put-function-concurrency` configura 100 execuções simultâneas reservadas para a função `my-function`.

```
aws lambda put-function-concurrency \
  --function-name my-function \
  --reserved-concurrent-executions 100
```

Saída:

```
{
  "ReservedConcurrentExecutions": 100
}
```

Para obter mais informações, consulte [Como reservar simultaneidade para uma função do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [PutFunctionConcurrency](#) na Referência de AWS CLI Comandos.

put-function-event-invoke-config

O código de exemplo a seguir mostra como usar `put-function-event-invoke-config`.

AWS CLI

Para configurar o tratamento de erros para invocação assíncrona

O `put-function-event-invoke-config` exemplo a seguir define a duração máxima do evento de uma hora e desativa as novas tentativas para a função especificada.

```
aws lambda put-function-event-invoke-config \  
  --function-name my-function \  
  --maximum-event-age-in-seconds 3600 \  
  --maximum-retry-attempts 0
```

Saída:

```
{  
  "LastModified": 1573686021.479,  
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:  
$LATEST",  
  "MaximumRetryAttempts": 0,  
  "MaximumEventAgeInSeconds": 3600,  
  "DestinationConfig": {  
    "OnSuccess": {},  
    "OnFailure": {}  
  }  
}
```

- Para API obter detalhes, consulte [PutFunctionEventInvokeConfig](#) na Referência de AWS CLI Comandos.

put-provisioned-concurrency-config

O código de exemplo a seguir mostra como usar `put-provisioned-concurrency-config`.

AWS CLI

Para alocar a simultaneidade provisionada

O seguinte exemplo de `put-provisioned-concurrency-config` aloca 100 simultaneidades provisionadas para o alias BLUE da função especificada.

```
aws lambda put-provisioned-concurrency-config \  
  --function-name my-function \  
  --provisioned-concurrency 100
```



```
--qualifier BLUE \  
--provisioned-concurrent-executions 100
```

Saída:

```
{  
  "Requested ProvisionedConcurrentExecutions": 100,  
  "Allocated ProvisionedConcurrentExecutions": 0,  
  "Status": "IN_PROGRESS",  
  "LastModified": "2019-11-21T19:32:12+0000"  
}
```

- Para API obter detalhes, consulte [PutProvisionedConcurrencyConfig](#) na Referência de AWS CLI Comandos.

remove-layer-version-permission

O código de exemplo a seguir mostra como usar `remove-layer-version-permission`.

AWS CLI

Para excluir permissões de versão de camada

O `remove-layer-version-permission` exemplo a seguir exclui a permissão de uma conta para configurar uma versão de camada.

```
aws lambda remove-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --version-number 1
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS Lambda Layers no Lambda Developer AWS Guide](#).

- Para API obter detalhes, consulte [RemoveLayerVersionPermission](#) na Referência de AWS CLI Comandos.

remove-permission

O código de exemplo a seguir mostra como usar `remove-permission`.

AWS CLI

Para remover permissões de uma função do Lambda existente

O seguinte exemplo de `remove-permission` remove permissão para invocar uma função chamada `my-function`.

```
aws lambda remove-permission \  
  --function-name my-function \  
  --statement-id sns
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando políticas baseadas em recursos para o Lambda AWS no Guia do desenvolvedor](#) do AWS Lambda.

- Para API obter detalhes, consulte [RemovePermission](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a uma função do Lambda existente

O seguinte exemplo de `tag-resource` adiciona uma tag com o nome de chave `DEPARTMENT` e um valor de `Department A` à função do Lambda especificada.

```
aws lambda tag-resource \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --tags "DEPARTMENT=Department A"
```

Este comando não produz saída.

Para obter mais informações, consulte [Como marcar funções do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de uma função do Lambda existente

O seguinte exemplo de `untag-resource` remove a tag com o nome de chave `DEPARTMENT` da função `my-function` do Lambda.

```
aws lambda untag-resource \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --tag-keys DEPARTMENT
```

Este comando não produz saída.

Para obter mais informações, consulte [Como marcar funções do Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-alias

O código de exemplo a seguir mostra como usar `update-alias`.

AWS CLI

Para atualizar um alias de função

O seguinte exemplo de `update-alias` atualiza um alias chamado `LIVE` que aponta para a versão 3 da função `my-function` do Lambda.

```
aws lambda update-alias \  
  --function-name my-function \  
  --function-version 3 \  
  --name LIVE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Configurando aliases de funções do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [UpdateAlias](#) na Referência de AWS CLI Comandos.

update-event-source-mapping

O código de exemplo a seguir mostra como usar `update-event-source-mapping`.

AWS CLI

Para atualizar o mapeamento entre uma fonte de eventos e uma função AWS Lambda

O `update-event-source-mapping` exemplo a seguir atualiza o tamanho do lote para 8 no mapeamento especificado.

```
aws lambda update-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --batch-size 8
```

Saída:

```
{  
  "UUID": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",  
  "StateTransitionReason": "USER_INITIATED",  
  "LastModified": 1569284520.333,  
  "BatchSize": 8,  
  "State": "Updating",  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "EventSourceArn": "arn:aws:sqs:us-west-2:123456789012:mySQSqueue"  
}
```

Para obter mais informações, consulte [AWS Lambda Event Source Mapping](#) no AWS Lambda Developer Guide.

- Para API obter detalhes, consulte [UpdateEventSourceMapping](#) na Referência de AWS CLI Comandos.

update-function-code

O código de exemplo a seguir mostra como usar `update-function-code`.

AWS CLI

Atualizar o código de uma função do Lambda

O `update-function-code` exemplo a seguir substitui o código da versão não publicada (\$LATEST) da `my-function` função pelo conteúdo do arquivo zip especificado.

```
aws lambda update-function-code \  
  --function-name my-function \  
  --zip-file fileb://my-function.zip
```

Saída:

```
{  
  "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-uy3l9qq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qqzYUHaqWR716pZ2dpkn/0J1FrzJmLKidWoaCgk=",  
  "Description": "",  
  "VpcConfig": {  
    "SubnetIds": [],  
    "VpcId": "",  
    "SecurityGroupIds": []  
  },  
  "CodeSize": 304,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "Handler": "index.handler"  
}
```

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [UpdateFunctionCode](#) na Referência de AWS CLI Comandos.

update-function-configuration

O código de exemplo a seguir mostra como usar `update-function-configuration`.

AWS CLI

Modificar a configuração de uma função

O `update-function-configuration` exemplo a seguir modifica o tamanho da memória para 256 MB para a versão não publicada (\$LATEST) da `my-function` função.

```
aws lambda update-function-configuration \  
  --function-name my-function \  
  --memory-size 256
```

Saída:

```
{  
  "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/0J1FrzJm1KidWoaCgk=",  
  "Description": "",  
  "VpcConfig": {  
    "SubnetIds": [],  
    "VpcId": "",  
    "SecurityGroupIds": []  
  },  
  "CodeSize": 304,  
  "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",  
  "Handler": "index.handler"  
}
```

Para obter mais informações, consulte [Configurar opções da função do AWS Lambda](#) no Guia do desenvolvedor do AWS Lambda.

- Para API obter detalhes, consulte [UpdateFunctionConfiguration](#) na Referência de AWS CLI Comandos.

update-function-event-invoke-config

O código de exemplo a seguir mostra como usar `update-function-event-invoke-config`.

AWS CLI

Para atualizar uma configuração de invocação assíncrona

O `update-function-event-invoke-config` exemplo a seguir adiciona um destino em caso de falha à configuração de invocação assíncrona existente para a função especificada.

```
aws lambda update-function-event-invoke-config \  
  --function-name my-function \  
  --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-  
east-2:123456789012:destination"}}'
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [UpdateFunctionEventInvokeConfig](#) na Referência de AWS CLI Comandos.

Exemplos de License Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with License Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-license-configuration

O código de exemplo a seguir mostra como usar `create-license-configuration`.

AWS CLI

Exemplo 1: Para criar uma configuração de licença

O `create-license-configuration` exemplo a seguir cria uma configuração de licença com um limite rígido de 10 núcleos.

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type Core \  
  --license-count 10 \  
  --license-count-hard-limit
```

Saída:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"  
}
```

Exemplo 2: Para criar uma configuração de licença

O `create-license-configuration` exemplo a seguir cria uma configuração de licença com um limite flexível de 100vCPUs. Ele usa uma regra para habilitar a CPU otimização v.

```
aws license-manager create-license-configuration --name my-license-configuration
--license-counting-type vCPU \
--license-count 100 \
--license-rules "#honorVcpuOptimization=true"
```

Saída:

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"
}
```

- Para API obter detalhes, consulte [CreateLicenseConfiguration](#) na Referência de AWS CLI Comandos.

delete-license-configuration

O código de exemplo a seguir mostra como usar `delete-license-configuration`.

AWS CLI

Para excluir uma configuração de licença

O `delete-license-configuration` exemplo a seguir exclui a configuração de licença especificada.

```
aws license-manager delete-license-configuration \
--license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteLicenseConfiguration](#) na Referência de AWS CLI Comandos.

get-license-configuration

O código de exemplo a seguir mostra como usar `get-license-configuration`.

AWS CLI

Para obter informações sobre a configuração da licença

O `get-license-configuration` exemplo a seguir exibe detalhes da configuração de licença especificada.

```
aws license-manager get-license-configuration \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Saída:

```
{  
  "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
}
]
```

- Para API obter detalhes, consulte [GetLicenseConfiguration](#) na Referência de AWS CLI Comandos.

get-service-settings

O código de exemplo a seguir mostra como usar `get-service-settings`.

AWS CLI

Para obter as configurações do License Manager

O `get-service-settings` exemplo a seguir exibe as configurações de serviço do License Manager na região atual.

```
aws license-manager get-service-settings
```

Veja a seguir um exemplo de saída se a descoberta de recursos entre contas estiver desativada.

```
{
  "OrganizationConfiguration": {
    "EnableIntegration": false
  },
  "EnableCrossAccountsDiscovery": false
}
```

Veja a seguir um exemplo de saída se a descoberta de recursos entre contas estiver ativada.

```
{
  "S3BucketArn": "arn:aws:s3::aws-license-manager-service-c22d6279-35c4-47c4-bb",
  "OrganizationConfiguration": {
    "EnableIntegration": true
  },
  "EnableCrossAccountsDiscovery": true
}
```

- Para API obter detalhes, consulte [GetServiceSettings](#) na Referência de AWS CLI Comandos.

list-associations-for-license-configuration

O código de exemplo a seguir mostra como usar `list-associations-for-license-configuration`.

AWS CLI

Para obter associações para uma configuração de licença

O `list-associations-for-license-configuration` exemplo a seguir exibe informações detalhadas sobre as associações da configuração de licença especificada.

```
aws license-manager list-associations-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Saída:

```
{
  "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
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [ListAssociationsForLicenseConfiguration](#) na Referência de AWS CLI Comandos.

list-license-configurations

O código de exemplo a seguir mostra como usar `list-license-configurations`.

AWS CLI

Exemplo 1: Para listar todas as suas configurações de licença

O `list-license-configurations` exemplo a seguir lista todas as suas configurações de licença.

```
aws license-manager list-license-configurations
```

Saída:

```
{
  "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
      }
    ]
  },
  {
    ...
  }
]
}

```

Exemplo 2: Para listar uma configuração de licença específica

O `list-license-configurations` exemplo a seguir lista somente a configuração de licença especificada.

```

aws license-manager list-license-configurations \
  --license-configuration-arns arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE

```

- Para API obter detalhes, consulte [ListLicenseConfigurations](#) na Referência de AWS CLI Comandos.

list-license-specifications-for-resource

O código de exemplo a seguir mostra como usar `list-license-specifications-for-resource`.

AWS CLI

Para listar as configurações de licença de um recurso

O `list-license-specifications-for-resource` exemplo a seguir lista as configurações de licença associadas à Amazon Machine Image (AMI) especificada.

```
aws license-manager list-license-specifications-for-resource \
  --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0
```

Saída:

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"
}
```

- Para API obter detalhes, consulte [ListLicenseSpecificationsForResource](#) na Referência de AWS CLI Comandos.

list-resource-inventory

O código de exemplo a seguir mostra como usar `list-resource-inventory`.

AWS CLI

Para listar recursos no inventário de recursos

O `list-resource-inventory` exemplo a seguir lista os recursos gerenciados usando o inventário do Systems Manager.

```
aws license-manager list-resource-inventory
```

Saída:

```
{
```

```
"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"
  }
]
```

- Para API obter detalhes, consulte [ListResourceInventory](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de uma configuração de licença

O `list-tags-for-resource` exemplo a seguir lista as tags para a configuração de licença especificada.


```
aws license-manager list-tags-for-resource \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-  
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Saída:

```
{  
  "Tags": [  
    {  
      "Key": "project",  
      "Value": "lima"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-usage-for-license-configuration

O código de exemplo a seguir mostra como usar `list-usage-for-license-configuration`.

AWS CLI

Para listar as licenças em uso para uma configuração de licença

O `list-usage-for-license-configuration` exemplo a seguir lista informações sobre os recursos que usam licenças para a configuração de licença especificada. Por exemplo, se o tipo de licença for vCPU, todas as instâncias consumirão uma licença por CPU v.

```
aws license-manager list-usage-for-license-configuration \  
  --license-configuration-arn arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Saída:

```
{  
  "LicenseConfigurationUsageList": [  
    {  
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/  
i-04a636d18e83cfac",
```

```
    "ResourceType": "EC2_INSTANCE",
    "ResourceStatus": "running",
    "ResourceOwnerId": "123456789012",
    "AssociationTime": 1570892850.519,
    "ConsumedLicenses": 2
  }
]
```

- Para API obter detalhes, consulte [ListUsageForLicenseConfiguration](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag, uma configuração de licença

O `tag-resource` exemplo a seguir adiciona a tag especificada (nome e valor da chave) à configuração da licença especificada.

```
aws license-manager tag-resource \
  --tags Key=project, Value=Lima \
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de uma configuração de licença

O `untag-resource` exemplo a seguir remove a tag especificada (nome da chave e recurso) da configuração de licença especificada.

```
aws license-manager untag-resource \  
  --tag-keys project \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-  
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-license-configuration

O código de exemplo a seguir mostra como usar update-license-configuration.

AWS CLI

Para atualizar uma configuração de licença

O update-license-configuration exemplo a seguir atualiza a configuração da licença especificada para remover o limite rígido.

```
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
```

Este comando não produz saída.

O update-license-configuration exemplo a seguir atualiza a configuração da licença especificada para alterar seu status para 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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateLicenseConfiguration](#) na Referência de AWS CLI Comandos.

update-license-specifications-for-resource

O código de exemplo a seguir mostra como usar `update-license-specifications-for-resource`.

AWS CLI

Para atualizar as configurações de licença de um recurso

O `update-license-specifications-for-resource` exemplo a seguir substitui a configuração da licença associada à Amazon Machine Image (AMI) especificada removendo uma configuração de licença e adicionando outra.

```
aws license-manager update-license-specifications-for-resource \  
  --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0 \  
  --remove-license-specifications LicenseConfigurationArn=arn:aws:license-  
manager:us-west-2:123456789012:license-  
configuration:lic-38b658717b87478aaa7c00883EXAMPLE \  
  --add-license-specifications LicenseConfigurationArn=arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-42b6deb06e5399a980d555927EXAMPLE
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateLicenseSpecificationsForResource](#) na Referência de AWS CLI Comandos.

update-service-settings

O código de exemplo a seguir mostra como usar `update-service-settings`.

AWS CLI

Para atualizar as configurações do License Manager

O `update-service-settings` exemplo a seguir permite a descoberta de recursos entre contas para o License Manager na AWS região atual. O bucket do Amazon S3 é a sincronização de dados de recursos necessária para o inventário do 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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateServiceSettings](#) na Referência de AWS CLI Comandos.

Exemplos do Lightsail usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Lightsail.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

allocate-static-ip

O código de exemplo a seguir mostra como usar `allocate-static-ip`.

AWS CLI

Para criar um IP estático

O `allocate-static-ip` exemplo a seguir cria o IP estático especificado, que pode ser anexado a uma instância.

```
aws lightsail allocate-static-ip \  
  --static-ip-name StaticIp-1
```

Saída:

```
{
```

```

"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
  }
]
}

```

- Para API obter detalhes, consulte [AllocateStaticIp](#) na Referência de AWS CLI Comandos.

attach-disk

O código de exemplo a seguir mostra como usar `attach-disk`.

AWS CLI

Para conectar um disco de armazenamento em bloco a uma instância

O `attach-disk` exemplo a seguir anexa disco à instância `WordPress_Multisite-1` com o caminho do disco de `/dev/xvdf`

```

aws lightsail attach-disk \
  --disk-name Disk-1 \
  --disk-path /dev/xvdf \
  --instance-name WordPress_Multisite-1

```

Saída:

```

{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [AttachDisk](#) na Referência de AWS CLI Comandos.

attach-instances-to-load-balancer

O código de exemplo a seguir mostra como usar `attach-instances-to-load-balancer`.

AWS CLI

Para anexar instâncias a um balanceador de carga

O `attach-instances-to-load-balancer` exemplo a seguir anexa instâncias `MEAN-1`, `MEAN-2`, e `MEAN-3` ao `LoadBalancer-1` balanceador de carga.

```
aws lightsail attach-instances-to-load-balancer \  
  --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \  
  --load-balancer-name LoadBalancer-1
```

Saída:

```
{  
  "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,  
      "status": "Started",  
      "statusChangedAt": 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
  }
]
}

```

- Para API obter detalhes, consulte [AttachInstancesToLoadBalancer](#) na Referência de AWS CLI Comandos.

attach-load-balancer-tls-certificate

O código de exemplo a seguir mostra como usar `attach-load-balancer-tls-certificate`.

AWS CLI

Para anexar um TLS certificado a um balanceador de carga

O `attach-load-balancer-tls-certificate` exemplo a seguir anexa o TLS certificado do balanceador de carga `Certificate2` ao balanceador de carga `LoadBalancer-1`.

```

aws lightsail attach-load-balancer-tls-certificate \
  --certificate-name Certificate2 \
  --load-balancer-name LoadBalancer-1

```

Saída:

```

{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [AttachLoadBalancerTlsCertificate](#) na Referência de AWS CLI Comandos.

attach-static-ip

O código de exemplo a seguir mostra como usar `attach-static-ip`.

AWS CLI

Para anexar um IP estático a uma instância

O `attach-static-ip` exemplo a seguir anexa IP estático `StaticIp-1` à instância `MEAN-1`.

```
aws lightsail attach-static-ip \
```

```
--static-ip-name StaticIp-1 \  
--instance-name MEAN-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [AttachStaticIp](#) na Referência de AWS CLI Comandos.

close-instance-public-ports

O código de exemplo a seguir mostra como usar `close-instance-public-ports`.

AWS CLI

Para fechar as portas de firewall de uma instância

O `close-instance-public-ports` exemplo a seguir fecha a TCP porta 22 na instância `MEAN-2`.

```
aws lightsail close-instance-public-ports \  
  --instance-name MEAN-2 \  
  --port-info fromPort=22,protocol=TCP,toPort=22
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [CloseInstancePublicPorts](#) na Referência de AWS CLI Comandos.

copy-snapshot

O código de exemplo a seguir mostra como usar `copy-snapshot`.

AWS CLI

Exemplo 1: Para copiar um instantâneo na mesma região AWS

O copy-snapshot exemplo a seguir copia o instantâneo da instância MEAN-1-1571075291 como instantâneo da instância MEAN-1-Copy na mesma AWS região. us-west-2

```
aws lightsail copy-snapshot \  
  --source-snapshot-name MEAN-1-1571075291 \  
  --target-snapshot-name MEAN-1-Copy \  
  --source-region us-west-2
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Cópia de snapshots de uma AWS região para outra no Amazon Lightsail no Lightsail Dev Guide](#).

Exemplo 2: Para copiar um instantâneo de uma AWS região para outra

O copy-snapshot exemplo a seguir copia o instantâneo da instância MEAN-1-1571075291 como instantâneo MEAN-1-1571075291-Copy da instância da AWS Região us-west-2 para 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
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Cópia de snapshots de uma AWS região para outra no Amazon Lightsail no Lightsail Dev Guide](#).

Exemplo 3: Para copiar um instantâneo automático na mesma região AWS

O copy-snapshot exemplo a seguir copia o instantâneo automático 2019-10-14 da instância WordPress-1 como um instantâneo manual WordPress-1-10142019 na AWS região. 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
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte Como [manter snapshots automáticos de instâncias ou discos no Amazon Lightsail no Lightsail Dev Guide](#).

Exemplo 4: Para copiar um instantâneo automático de uma AWS região para outra

O copy-snapshot exemplo a seguir copia o instantâneo automático 2019-10-14 da instância WordPress-1 como um instantâneo manual WordPress-1-10142019 da AWS Região us-west-2 para 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
```

Saída:

```
{
```



```
"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  
  }  
]  
}
```

Para obter mais informações, consulte Como [manter snapshots automáticos de instâncias ou discos no Amazon Lightsail no Lightsail Dev Guide](#).

- Para API obter detalhes, consulte [CopySnapshot](#) na Referência de AWS CLI Comandos.

create-disk-from-snapshot

O código de exemplo a seguir mostra como usar `create-disk-from-snapshot`.

AWS CLI

Para criar e criar um disco a partir de um instantâneo de disco

O `create-disk-from-snapshot` exemplo a seguir cria um disco de armazenamento em bloco chamado `Disk-2` partir do instantâneo do disco de armazenamento em bloco especificado. O disco é criado na AWS região e zona de disponibilidade especificadas, com 32 GB de espaço de armazenamento.

```
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
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Criação de um disco de armazenamento em bloco a partir de um snapshot no Amazon Lightsail no Guia do desenvolvedor do Lightsail](#).

- Para API obter detalhes, consulte [CreateDiskFromSnapshot](#) na Referência de AWS CLI Comandos.

create-disk-snapshot

O código de exemplo a seguir mostra como usar `create-disk-snapshot`.

AWS CLI

Exemplo 1: Para criar um instantâneo de um disco

O `create-disk-snapshot` exemplo a seguir cria um instantâneo chamado `DiskSnapshot-1` do disco de armazenamento em bloco especificado.

```
aws lightsail create-disk-snapshot \
  --disk-name Disk-1 \
  --disk-snapshot-name DiskSnapshot-1
```

Saída:

```
{
  "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
    }
  ]
}
```

Exemplo 2: Para criar um snapshot do disco do sistema de uma instância

O `create-disk-snapshot` exemplo a seguir cria um instantâneo do disco do sistema da instância especificada.

```
aws lightsail create-disk-snapshot \
```

```
--instance-name WordPress-1 \  
--disk-snapshot-name SystemDiskSnapshot-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Snapshots no Amazon Lightsail e Criação de um snapshot do volume raiz de uma instância no Amazon Lightsail no Guia do desenvolvedor do Lightsail](#).

- Para API obter detalhes, consulte [CreateDiskSnapshot](#) na Referência de AWS CLI Comandos.

create-disk

O código de exemplo a seguir mostra como usar create-disk.

AWS CLI

Para criar um disco de armazenamento em bloco

O create-disk exemplo a seguir cria um disco de armazenamento Disk-1 em bloco na AWS região e zona de disponibilidade especificadas, com 32 GB de espaço de armazenamento.

```
aws lightsail create-disk \  
  --disk-name Disk-1 \  
  --availability-zone us-west-2a \  
  --size-in-gb 32
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [CreateDisk](#) na Referência de AWS CLI Comandos.

create-domain-entry

O código de exemplo a seguir mostra como usar `create-domain-entry`.

AWS CLI

Para criar uma entrada de domínio (DNSregistro)

O `create-domain-entry` exemplo a seguir cria um DNS registro (A) para o ápice do domínio especificado que aponta para o endereço IP de uma instância.

Observação: as API operações relacionadas ao domínio do Lightsail estão disponíveis somente na região. `us-east-1` Se seu CLI perfil estiver configurado para usar uma região diferente, você deverá incluir o `--region us-east-1` parâmetro ou o comando falhará.

```
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
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [DNSno Amazon Lightsail e Como criar DNS uma zona para gerenciar os registros do seu domínio no Amazon Lightsail no Guia DNS do desenvolvedor do Lightsail](#).

- Para API obter detalhes, consulte [CreateDomainEntry](#) na Referência de AWS CLI Comandos.

create-domain

O código de exemplo a seguir mostra como usar `create-domain`.

AWS CLI

Para criar um domínio (DNSzona)

O `create-domain` exemplo a seguir cria uma DNS zona para o domínio especificado.

Observação: as API operações relacionadas ao domínio do Lightsail estão disponíveis somente na região. `us-east-1` Se seu CLI perfil estiver configurado para usar uma região diferente, você deverá incluir o `--region us-east-1` parâmetro ou o comando falhará.

```
aws lightsail create-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [DNS no Amazon Lightsail e Como criar DNS uma zona para gerenciar os registros do seu domínio no Amazon Lightsail no Guia DNS do desenvolvedor do Lightsail](#).

- Para API obter detalhes, consulte [CreateDomain](#) na Referência de AWS CLI Comandos.

create-instance-snapshot

O código de exemplo a seguir mostra como usar `create-instance-snapshot`.

AWS CLI

Para criar um instantâneo de uma instância

O `create-instance-snapshot` exemplo a seguir cria um instantâneo da instância especificada.

```
aws lightsail create-instance-snapshot \  
  --instance-name WordPress-1 \  
  --instance-snapshot-name WordPress-Snapshot-1
```

Saída:

```
{  
  "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
    }
  ]
}

```

- Para API obter detalhes, consulte [CreateInstanceSnapshot](#) na Referência de AWS CLI Comandos.

create-instances-from-snapshot

O código de exemplo a seguir mostra como usar `create-instances-from-snapshot`.

AWS CLI

Para criar uma instância a partir de um snapshot

O `create-instances-from-snapshot` exemplo a seguir cria uma instância a partir do instantâneo da instância especificada, na AWS região e zona de disponibilidade especificadas, usando o pacote de \$10USD.

Observação: o pacote que você especifica deve ser igual ou maior em especificações do que o pacote da instância de origem original usado para criar o snapshot.

```

aws lightsail create-instances-from-snapshot \
  --instance-snapshot-name WordPress-1-1569866208 \
  --instance-names WordPress-2 \
  --availability-zone us-west-2a \
  --bundle-id medium_2_0

```

Saída:

```

{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [CreateInstancesFromSnapshot](#) na Referência de AWS CLI Comandos.

create-instances

O código de exemplo a seguir mostra como usar `create-instances`.

AWS CLI

Exemplo 1: Para criar uma única instância

O `create-instances` exemplo a seguir cria uma instância na AWS região e zona de disponibilidade especificadas, usando o WordPress blueprint e o pacote de 3,50 USDUSD.

```

aws lightsail create-instances \
  --instance-names Instance-1 \
  --availability-zone us-west-2a \
  --blueprint-id wordpress_5_1_1_2 \
  --bundle-id nano_2_0

```

Saída:

```

{
  "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
    }
]
}

```

Exemplo 2: Para criar várias instâncias ao mesmo tempo

O `create-instances` exemplo a seguir cria três instâncias na AWS região e zona de disponibilidade especificadas, usando o WordPress blueprint e o pacote de 3,50 USDUSD.

```

aws lightsail create-instances \
  --instance-names {"Instance1","Instance2","Instance3"} \
  --availability-zone us-west-2a \
  --blueprint-id wordpress_5_1_1_2 \
  --bundle-id nano_2_0

```

Saída:

```

{
  "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
  }
]
}
```

- Para API obter detalhes, consulte [CreateInstances](#) na Referência de AWS CLI Comandos.

create-key-pair

O código de exemplo a seguir mostra como usar `create-key-pair`.

AWS CLI

Para criar um par de chaves

O `create-key-pair` exemplo a seguir cria um par de chaves que você pode usar para autenticar e se conectar a uma instância.

```
aws lightsail create-key-pair \
  --key-pair-name MyPersonalKeyPair
```

A saída fornece o valor base64 da chave privada que você pode usar para autenticar instâncias que usam o par de chaves criado. Observação: copie e cole o valor base64 da chave privada em um local seguro porque você não pode recuperá-lo posteriormente.

```
{
  "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
+BUUgedGUXno6uF7agqxZN01kPLJBIVTW26SSYBJ0tE
+y804UyVsjrUqCaMXDhmfXpWuLMPwuXhwckh7e8hwoTfkiX0E6Q1
+KqF/MiA3w6DCjEqvvdI07SiEZJFsuGNfYDDN3w60Rel5MUhmn30Jdn4y/
A7Nwb3IxL4pPvFE4rgFRKU8n1jp9kwRnLVMVB0WuGXk6n+H6M2f1 ",
  "privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCaFICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6\nnb24xFDASBgNVBAAsTC01BTSBD
\nBkgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
\nVQQHEwdTZWF0dGx1MQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBAAsTC01BTSBDb25z
\nb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
\nYXpvbi5jb20wZGZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
\nIbb30hjZnzcVQAaREXAMPLEm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEATCu4\nnnUHVvXyUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780EXAMPLELvjx79LjStB
\nNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=\n-----END RSA PRIVATE KEY-----",
  "operation": {
    "id": "67f984db-9994-45fe-ad38-59bafcaf82ef",
```

```

    "resourceName": "MyPersonalKeyPair",
    "resourceType": "KeyPair",
    "createdAt": 1569866556.567,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationType": "CreateKeyPair",
    "status": "Succeeded",
    "statusChangedAt": 1569866556.704
  }
}

```

- Para API obter detalhes, consulte [CreateKeyPair](#) na Referência de AWS CLI Comandos.

create-load-balancer-tls-certificate

O código de exemplo a seguir mostra como usar `create-load-balancer-tls-certificate`.

AWS CLI

Para criar um TLS certificado para um balanceador de carga

O `create-load-balancer-tls-certificate` exemplo a seguir cria um TLS certificado anexado ao balanceador de carga especificado. O certificado criado se aplica aos domínios especificados. Observação: somente dois certificados podem ser criados para um balanceador de carga.

```

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

```

Saída:

```

{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [CreateLoadBalancerTlsCertificate](#) na Referência de AWS CLI Comandos.

create-load-balancer

O código de exemplo a seguir mostra como usar create-load-balancer.

AWS CLI

Para criar um balanceador de carga

O `create-load-balancer` exemplo a seguir cria um balanceador de carga com um TLS certificado. O TLS certificado se aplica aos domínios especificados e encaminha o tráfego para instâncias na 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
```

Saída:

```
{  
  "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
    }
  ]
}
```

Para obter mais informações, consulte Balanceadores de [carga do Lightsail](#) no Guia do desenvolvedor do Lightsail.

- Para API obter detalhes, consulte [CreateLoadBalancer](#) na Referência de AWS CLI Comandos.

create-relational-database-from-snapshot

O código de exemplo a seguir mostra como usar `create-relational-database-from-snapshot`.

AWS CLI

Para criar um banco de dados gerenciado a partir de um snapshot

O `create-relational-database-from-snapshot` exemplo a seguir cria um banco de dados gerenciado a partir do snapshot especificado na AWS região e zona de disponibilidade especificadas, usando o pacote de banco de dados USD padrão de \$15. Observação: O pacote que você especifica deve ser igual ou maior em especificações do que o pacote do banco de dados de origem original usado para criar o snapshot.

```
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
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [CreateRelationalDatabaseFromSnapshot](#) na Referência de AWS CLI Comandos.

create-relational-database-snapshot

O código de exemplo a seguir mostra como usar `create-relational-database-snapshot`.

AWS CLI

Para criar um instantâneo de um banco de dados gerenciado

O `create-relational-database-snapshot` exemplo a seguir cria um instantâneo do banco de dados gerenciado especificado.

```
aws lightsail create-relational-database-snapshot \  
--relational-database-name Database1 \  
--no-publicly-accessible
```

```
--relational-database-snapshot-name RelationalDatabaseSnapshot1
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [CreateRelationalDatabaseSnapshot](#) na Referência de AWS CLI Comandos.

create-relational-database

O código de exemplo a seguir mostra como usar `create-relational-database`.

AWS CLI

Para criar um banco de dados gerenciado

O `create-relational-database` exemplo a seguir cria um banco de dados gerenciado na AWS região e zona de disponibilidade especificadas, usando o mecanismo de banco de dados My SQL 5.6 (`mysql_5_6`) e o pacote de banco de dados padrão de USD \$15 (`micro_1_0`). O banco de dados gerenciado é pré-preenchido com um nome de usuário principal e não pode ser acessado publicamente.

```
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
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

```
}
```

- Para API obter detalhes, consulte [CreateRelationalDatabase](#) na Referência de AWS CLI Comandos.

delete-auto-snapshot

O código de exemplo a seguir mostra como usar delete-auto-snapshot.

AWS CLI

Para excluir um instantâneo automático

O delete-auto-snapshot exemplo a seguir exclui o instantâneo automático 2019-10-10 da instância WordPress-1

```
aws lightsail delete-auto-snapshot \  
  --resource-name WordPress-1 \  
  --date 2019-10-10
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Excluir snapshots automáticos de instâncias ou discos no Amazon Lightsail no Guia do desenvolvedor do Lightsail](#).

- Para API obter detalhes, consulte [DeleteAutoSnapshot](#) na Referência de AWS CLI Comandos.

delete-disk-snapshot

O código de exemplo a seguir mostra como usar `delete-disk-snapshot`.

AWS CLI

Para excluir um instantâneo de um disco de armazenamento em bloco

O `delete-disk-snapshot` exemplo a seguir exclui o instantâneo especificado de um disco de armazenamento em bloco.

```
aws lightsail delete-disk-snapshot \  
  --disk-snapshot-name DiskSnapshot-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DeleteDiskSnapshot](#) na Referência de AWS CLI Comandos.

delete-disk

O código de exemplo a seguir mostra como usar delete-disk.

AWS CLI

Para excluir um disco de armazenamento em bloco

O delete-disk exemplo a seguir exclui o disco de armazenamento em bloco especificado.

```
aws lightsail delete-disk \  
  --disk-name Disk-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DeleteDisk](#) na Referência de AWS CLI Comandos.

delete-domain-entry

O código de exemplo a seguir mostra como usar delete-domain-entry.

AWS CLI

Para excluir uma entrada de domínio (DNSregistro)

O `delete-domain-entry` exemplo a seguir exclui a entrada de domínio especificada de um domínio existente.

Observação: as API operações relacionadas ao domínio do Lightsail estão disponíveis somente na região `us-east-1`. Se seu CLI perfil estiver configurado para usar uma região diferente, você deverá incluir o `--region us-east-1` parâmetro ou o comando falhará.

```
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
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [DeleteDomainEntry](#) na Referência de AWS CLI Comandos.

delete-domain

O código de exemplo a seguir mostra como usar `delete-domain`.

AWS CLI

Para excluir um domínio (DNSzona)

O `delete-domain` exemplo a seguir exclui o domínio especificado e todas as entradas no domínio (DNSregistros).

Observação: as API operações relacionadas ao domínio do Lightsail estão disponíveis somente na região. `us-east-1` Se seu CLI perfil estiver configurado para usar uma região diferente, você deverá incluir o `--region us-east-1` parâmetro ou o comando falhará.

```
aws lightsail delete-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [DeleteDomain](#) na Referência de AWS CLI Comandos.

delete-instance-snapshot

O código de exemplo a seguir mostra como usar `delete-instance-snapshot`.

AWS CLI

title

O `delete-instance-snapshot` exemplo a seguir exclui o instantâneo especificado de uma instância.

```
aws lightsail delete-instance-snapshot \  
  --instance-snapshot-name WordPress-1-Snapshot-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DeleteInstanceSnapshot](#) na Referência de AWS CLI Comandos.

delete-instance

O código de exemplo a seguir mostra como usar `delete-instance`.

AWS CLI

Para excluir uma instância

O `delete-instance` exemplo a seguir exclui a instância especificada.

```
aws lightsail delete-instance \  
  --instance-name WordPress-1
```

Saída:

```
{
  "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
      }
    ]
  }
```

- Para API obter detalhes, consulte [DeleteInstance](#) na Referência de AWS CLI Comandos.

delete-key-pair

O código de exemplo a seguir mostra como usar `delete-key-pair`.

AWS CLI

Para excluir um par de chaves

O `delete-key-pair` exemplo a seguir exclui o par de chaves especificado.

```
aws lightsail delete-key-pair \
  --key-pair-name MyPersonalKeyPair
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [DeleteKeyPair](#) na Referência de AWS CLI Comandos.

delete-known-host-keys

O código de exemplo a seguir mostra como usar `delete-known-host-keys`.

AWS CLI

Para excluir chaves de host conhecidas de uma instância

O `delete-known-host-keys` exemplo a seguir exclui a chave de host conhecida da instância especificada.

```
aws lightsail delete-known-host-keys \  
  --instance-name Instance-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Solução de problemas de conexão com o Amazon Lightsail baseado em SSH navegador ou RDP cliente no Guia do desenvolvedor do Lightsail](#).

- Para API obter detalhes, consulte [DeleteKnownHostKeys](#) na Referência de AWS CLI Comandos.

delete-load-balancer-tls-certificate

O código de exemplo a seguir mostra como usar `delete-load-balancer-tls-certificate`.

AWS CLI

Para excluir um TLS certificado de um balanceador de carga

O `delete-load-balancer-tls-certificate` exemplo a seguir exclui o TLS certificado especificado do balanceador de carga especificado.

```
aws lightsail delete-load-balancer-tls-certificate \  
  --load-balancer-name MyFirstLoadBalancer \  
  --certificate-name MyFirstCertificate
```

Saída:

```
{  
  "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": "CreateLoadBalancer",  
      "status": "Completed",  
      "statusChangedAt": 1569874989.48  
    }  
  ]  
}
```

```

        "operationType": "DeleteLoadBalancerTlsCertificate",
        "status": "Started",
        "statusChangedAt": 1569874989.48
    }
]
}

```

- Para API obter detalhes, consulte [DeleteLoadBalancerTlsCertificate](#) na Referência de AWS CLI Comandos.

delete-load-balancer

O código de exemplo a seguir mostra como usar `delete-load-balancer`.

AWS CLI

Como excluir um balanceador de carga

O `delete-load-balancer` exemplo a seguir exclui o balanceador de carga especificado e todos os certificados associados TLS.

```

aws lightsail delete-load-balancer \
  --load-balancer-name MyFirstLoadBalancer

```

Saída:

```

{
  "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
    }
  ]
}

```

Para obter mais informações, consulte o título no guia.

- Para API obter detalhes, consulte [DeleteLoadBalancer](#) na Referência de AWS CLI Comandos.

delete-relational-database-snapshot

O código de exemplo a seguir mostra como usar `delete-relational-database-snapshot`.

AWS CLI

Para excluir um instantâneo de um banco de dados gerenciado

O `delete-relational-database-snapshot` exemplo a seguir exclui o instantâneo especificado de um banco de dados gerenciado.

```
aws lightsail delete-relational-database-snapshot \  
--relational-database-snapshot-name Database-Oregon-1-1566839359
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DeleteRelationalDatabaseSnapshot](#) na Referência de AWS CLI Comandos.

delete-relational-database

O código de exemplo a seguir mostra como usar `delete-relational-database`.

AWS CLI

Para excluir um banco de dados gerenciado

O `delete-relational-database` exemplo a seguir exclui o banco de dados gerenciado especificado.

```
aws lightsail delete-relational-database \  

```

```
--relational-database-name Database-1
```

Saída:

```
{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [DeleteRelationalDatabase](#) na Referência de AWS CLI Comandos.

detach-static-ip

O código de exemplo a seguir mostra como usar `detach-static-ip`.

AWS CLI

Para separar um IP estático de uma instância

O `detach-static-ip` exemplo a seguir separa o IP estático `StaticIp-1` de qualquer instância anexada.

```

aws lightsail detach-static-ip \
  --static-ip-name StaticIp-1

```

Saída:

```

{
  "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
  }
]
```

- Para API obter detalhes, consulte [DetachStaticIp](#) na Referência de AWS CLI Comandos.

get-active-names

O código de exemplo a seguir mostra como usar `get-active-names`.

AWS CLI

Para obter nomes de recursos ativos

O `get-active-names` exemplo a seguir retorna os nomes dos recursos ativos na AWS região configurada.

```
aws lightsail get-active-names
```

Saída:

```
{
  "activeNames": [
    "WordPress-1",
```

```
    "StaticIp-1",
    "MEAN-1",
    "Plesk_Hosting_Stack_on_Ubuntu-1"
  ]
}
```

- Para API obter detalhes, consulte [GetActiveNames](#) na Referência de AWS CLI Comandos.

get-auto-snapshots

O código de exemplo a seguir mostra como usar `get-auto-snapshots`.

AWS CLI

Para obter os instantâneos automáticos disponíveis para uma instância

O `get-auto-snapshots` exemplo a seguir retorna os instantâneos automáticos disponíveis, por exemplo `WordPress-1`.

```
aws lightsail get-auto-snapshots \
  --resource-name WordPress-1
```

Saída:

```
{
  "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": []  
    }  
]  
}
```

Para obter mais informações, consulte Como [manter snapshots automáticos de instâncias ou discos no Amazon Lightsail no Lightsail Dev Guide](#).

- Para API obter detalhes, consulte [GetAutoSnapshots](#) na Referência de AWS CLI Comandos.

get-blueprints

O código de exemplo a seguir mostra como usar `get-blueprints`.

AWS CLI

Para obter os planos para novas instâncias

O `get-blueprints` exemplo a seguir mostra detalhes sobre todos os esquemas disponíveis que podem ser usados para criar novas instâncias no Amazon Lightsail.

```
aws lightsail get-blueprints
```

Saída:

```
{  
  "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": "5.2.2-3",
    "versionCode": "1",
    "productUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U",
    "licenseUrl": "https://d7umqicpi7263.cloudfront.net/eula/
product/7d426cb7-9522-4dd7-a56b-55dd8cc1c8d0/588fd495-6492-4610-b3e8-
d15ce864454c.txt",
    "platform": "LINUX_UNIX"
  },
  {
    "blueprintId": "lamp_7_1_28",
    "name": "LAMP (PHP 7)",
    "group": "lamp_7",
    "type": "app",
    "description": "LAMP with PHP 7.x certified by Bitnami greatly
simplifies the development and deployment of PHP applications. It includes the
latest versions of PHP 7.x, Apache and MySQL together with phpMyAdmin and popular
PHP frameworks Zend, Symfony, CodeIgniter, CakePHP, Smarty, and Laravel. Other pre-
configured components and PHP modules include FastCGI, ModSecurity, SQLite, Varnish,
ImageMagick, xDebug, Xcache, OpenLDAP, Memcache, OAuth, PEAR, PECL, APC, GD and
cURL. It is secure by default and supports multiple applications, each with its own
virtual host and project directory. Let's Encrypt SSL certificates are supported
through an auto-configuration script.",
    "isActive": true,
    "minPower": 0,
    "version": "7.1.28",
    "versionCode": "1",
    "productUrl": "https://aws.amazon.com/marketplace/pp/B072JNJZ5C",
    "licenseUrl": "https://d7umqicpi7263.cloudfront.net/eula/product/
cb6afd05-a3b2-4916-a3e6-bccd414f5f21/12ab56cc-6a8c-4977-9611-dcd770824aad.txt",
    "platform": "LINUX_UNIX"
  },
  {
    "blueprintId": "nodejs",
    "name": "Node.js",

```

```

        "group": "node",
        "type": "app",
        "description": "Node.js certified 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": "12.7.0",
        "versionCode": "1",
        "productUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAK0",
        "licenseUrl": "https://d7umqicpi7263.cloudfront.net/
eula/product/033793fe-951d-47d0-aa94-5fbd0afb3582/25f8fa66-c868-4d80-
adf8-4a2b602064ae.txt",
        "platform": "LINUX_UNIX"
    },
    ...
}
]
}

```

- Para API obter detalhes, consulte [GetBlueprints](#) na Referência de AWS CLI Comandos.

get-bundles

O código de exemplo a seguir mostra como usar `get-bundles`.

AWS CLI

Para obter os pacotes para novas instâncias

O `get-bundles` exemplo a seguir mostra detalhes sobre todos os pacotes disponíveis que podem ser usados para criar novas instâncias no Amazon Lightsail.

```
aws lightsail get-bundles
```

Saída:


```
{
  "bundles": [
    {
      "price": 3.5,
      "cpuCount": 1,
      "diskSizeInGb": 20,
      "bundleId": "nano_2_0",
      "instanceType": "nano",
      "isActive": true,
      "name": "Nano",
      "power": 300,
      "ramSizeInGb": 0.5,
      "transferPerMonthInGb": 1024,
      "supportedPlatforms": [
        "LINUX_UNIX"
      ]
    },
    {
      "price": 5.0,
      "cpuCount": 1,
      "diskSizeInGb": 40,
      "bundleId": "micro_2_0",
      "instanceType": "micro",
      "isActive": true,
      "name": "Micro",
      "power": 500,
      "ramSizeInGb": 1.0,
      "transferPerMonthInGb": 2048,
      "supportedPlatforms": [
        "LINUX_UNIX"
      ]
    },
    {
      "price": 10.0,
      "cpuCount": 1,
      "diskSizeInGb": 60,
      "bundleId": "small_2_0",
      "instanceType": "small",
      "isActive": true,
      "name": "Small",
      "power": 1000,
      "ramSizeInGb": 2.0,
      "transferPerMonthInGb": 3072,
    }
  ]
}
```

```

        "supportedPlatforms": [
            "LINUX_UNIX"
        ]
    },
    ...
}
]
}

```

- Para API obter detalhes, consulte [GetBundles](#) na Referência de AWS CLI Comandos.

get-cloud-formation-stack-records

O código de exemplo a seguir mostra como usar `get-cloud-formation-stack-records`.

AWS CLI

Para obter os registros da CloudFormation pilha e suas pilhas associadas

O `get-cloud-formation-stack-records` exemplo a seguir mostra detalhes sobre os registros da CloudFormation pilha e suas pilhas associadas usadas para criar EC2 recursos da Amazon a partir de snapshots exportados do Amazon Lightsail.

```
aws lightsail get-cloud-formation-stack-records
```

Saída:

```

{
  "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"
    }
  }
]
}

```

- Para API obter detalhes, consulte [GetCloudFormationStackRecords](#) na Referência de AWS CLI Comandos.

get-disk-snapshot

O código de exemplo a seguir mostra como usar `get-disk-snapshot`.

AWS CLI

Para obter informações sobre um instantâneo de disco

O `get-disk-snapshot` exemplo a seguir exibe detalhes sobre o instantâneo `Disk-1-1566839161` do disco.

```

aws lightsail get-disk-snapshot \
  --disk-snapshot-name Disk-1-1566839161

```

Saída:

```

{
  "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
  }
}
```

Para obter mais informações, consulte o título no guia.

- Para API obter detalhes, consulte [GetDiskSnapshot](#) na Referência de AWS CLI Comandos.

get-disk-snapshots

O código de exemplo a seguir mostra como usar `get-disk-snapshots`.

AWS CLI

Para obter informações sobre todos os instantâneos de disco

O `get-disk-snapshots` exemplo a seguir exibe detalhes sobre todos os instantâneos de disco na AWS região configurada.

```
aws lightsail get-disk-snapshots
```

Saída:

```
{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [GetDiskSnapshots](#) na Referência de AWS CLI Comandos.

get-disk

O código de exemplo a seguir mostra como usar `get-disk`.

AWS CLI

Para obter informações sobre um disco de armazenamento em bloco

O `get-disk` exemplo a seguir exibe detalhes sobre o disco `Disk-1`.

```
aws lightsail get-disk \  
  --disk-name Disk-1
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte o título no guia.

- Para API obter detalhes, consulte [GetDisk](#) na Referência de AWS CLI Comandos.

get-disks

O código de exemplo a seguir mostra como usar `get-disks`.

AWS CLI

Para obter informações sobre todos os discos de armazenamento em bloco

O `get-disks` exemplo a seguir exibe detalhes sobre todos os discos na AWS região configurada.

```
aws lightsail get-disks
```

Saída:

```
{
  "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"
}
]
}

```

- Para API obter detalhes, consulte [GetDisks](#) na Referência de AWS CLI Comandos.

get-domain

O código de exemplo a seguir mostra como usar `get-domain`.

AWS CLI

Para obter informações sobre um domínio

O `get-domain` exemplo a seguir exibe detalhes sobre o domínio `example.com`.

Observação: as API operações relacionadas ao domínio do Lightsail estão disponíveis somente na região `us-east-1` AWS. Se seu CLI perfil estiver configurado para usar uma região diferente, você deverá incluir o parâmetro `--region us-east-1` ou o comando falhará.

```

aws lightsail get-domain \
  --domain-name example.com \
  --region us-east-1

```

Saída:

```

{
  "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"
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [GetDomain](#) na Referência de AWS CLI Comandos.

get-domains

O código de exemplo a seguir mostra como usar `get-domains`.

AWS CLI

Para obter informações sobre todos os domínios

O `get-domains` exemplo a seguir exibe detalhes sobre todos os domínios na AWS região configurada.

Observação: as API operações relacionadas ao domínio do Lightsail estão disponíveis somente na região. `us-east-1` AWS Se seu CLI perfil estiver configurado para usar uma região diferente, você deverá incluir o `--region us-east-1` parâmetro ou o comando falhará.

```
aws lightsail get-domains \
  --region us-east-1
```

Saída:

```

{
  "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.hkvuijqjou.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"
    }
]
}

```

- Para API obter detalhes, consulte [GetDomains](#) na Referência de AWS CLI Comandos.

get-export-snapshot-record

O código de exemplo a seguir mostra como usar `get-export-snapshot-record`.

AWS CLI

Para obter os registros de instantâneos exportados para a Amazon EC2

O `get-export-snapshot-record` exemplo a seguir mostra detalhes sobre a instância do Amazon Lightsail ou snapshots de disco exportados para a Amazon. EC2

```
aws lightsail get-export-snapshot-records
```

Saída:

```

{
  "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-Oregon-1-1540339219",
      "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/5446f534-ed60-4c17-b4a5-bEXAMPLEf8b7",
      "fromResourceName": "WordPress-512MB-Oregon-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"
    }
  }
]
}

```

- Para API obter detalhes, consulte [GetExportSnapshotRecord](#) na Referência de AWS CLI Comandos.

get-instance-access-details

O código de exemplo a seguir mostra como usar `get-instance-access-details`.

AWS CLI

Para obter informações sobre a chave do host de uma instância

O `get-instance-access-details` exemplo a seguir exibe informações da chave do host, por exemplo `WordPress_Multisite-1`.

```
aws lightsail get-instance-access-details \
  --instance-name WordPress_Multisite-1
```

Saída:

```
{
  "accessDetails": {
    "certKey": "ssh-rsa-cert-v01@openssh.com
AEXAMPLEEaC1yc2EtY2VydC12MDFAb3B1bnNzaC5jb20AAAAgNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
vEXAMPLE2eBWJyQvn7o1/
i0+s966h5sx8qUD791PB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjb1Lj26Fb37EKda9HVfC0u8pWbvky7Tyn9w29
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/
K1EXAMPLEAAAAAAAAABAAAALS00MzMzMDU4MzA40Dg1MTY2NjM40np6UWlndHk4UE1RSG9Stit0TG5QSEE9PQAAAAAASAA
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN
+tiXm7Y80gziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLUv7cnWPB7iU1HQg
+E3LUKrv4ZFw9pj7X2dFdNKfMxwWgI1ISWKimEXAMPLEEeHjrf1Rqc/
QH6TpWCvPfcx8uvwVqdwTfke/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/
KwmM7pCwJbSLDM07n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEEobqfdtyYJHHe11EyyEJs1fWNU3D5JIGlgzcPAV
+Z1bQyUCZXf0oslSa+HE85f0/
FRq9SVSBSHrmbeb0fr1PhgMzgSmqLeyhlbr6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPwvRgcr0qI2EL0tAajnpQQ8UZo
Aqter0xN5PhFL0J490WTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8c111327EXAMPLE/
e89GC89KcmKCxfGQniDAUgF8UqofIbq3Z0UgiAAYCVXc1I4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/
OzcTR+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
\nrEXAMPLEe08hJmqz2KFQ09X7fB21BruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaF10GNGQ8qy1HGUiscquZ9NDIL49n4mXbfsTH
\n0EXAMPLE12ZqsflYnSaUYCwjE74qH8ECVpytQIDAQABAoIBAHeZV9Z58JHAjifz
\nCEXAMPLEEqC3do0VDgXS1kKI92qNo4z2VcUEho878paCuVvXVHcCGgSnGeyIh2tN
\nMEXAMPLESohR427BhH3YLA+3Z5SivnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn
\nIEXAMPLEM93oF9eSZCjcLKB/jGhsfb0eCDMP8BshHE2beuqzVMoK1Dx0nvoP3+Fp
```

```

\nAEXAMPLESq6pDpCo9YVUX8g1u3Ro9cP12LXHDy+oVEY5KhbZQJ7VU1I72W0vppWW
\n0EXAMPLEkgYlq7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoNONQ9ZPaWHyJvymeud
\noQTNuz0CgYEA/LFWNTEZrzdzdR1kJmyNRmAermU0B6utyNENChAlHGSHkB+11VSh
\nbEXAMPLEQo9ooUeW5Ux03YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob
\n817CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/E0rB0we0RKyDSzWScGgYEA+z/r
\niob+nJZq0Ybn0SuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJd0NKDdkui0d52aAL
\nVgn7nLo88rVWKhJwVc4tu/rNgZLcR3bP4+kL6zand0KQnMLy0zNA2Ys26aa5udH1\nqw10WTt9WEm/
h10ndC1kn0MectrvsG17b38y5sMCgYEA54NiRGGz8oCPW6GN/FZA
\nKEXAMPLE5tw34GEH3UxlC9n3CejDaQmcz0ATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkc8Z71L/
agZEAaVCEog9FqfSqwB
+XTfoKh8qur74X1yCu9p6gof1q6k9\nEXAMPLEchJcNN0g4ETIfMkCgYBdVORRHE4mqvWp0dzA7v66FdEz2YSkjAXKK
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\nngEXAMPLEBogjw4MqHKL1bPKMHYQU8/
q24PaYgzHPzy13wLH6pTYf1XqlHdE2D6Vv\nnyEXAMPLEgQC3i/
kVVhky/2XRwRV1C7J02Bg3QGTx38hpmDa5IuofKANjA+wa3/zy\nnbEXAMPLE6ytQgD9GN/YtBq+uh0
+2ZkvXPL+CWRi0ZRXpPwYDBBFU9Cw0AuWWG1L8\nnwEXAMPLExMlcysRgcWB9RNgf3Au0pFd2i6XT/
riNsvvKpmJ+VooU8g==\n-----END RSA PRIVATE KEY-----\n",
    "protocol": "ssh",
    "instanceName": "WordPress_Multisite-1",
    "username": "bitnami",
    "hostKeys": [
        {
            "algorithm": "ssh-rsa",
            "publicKey":
"AEXAMPLEaC1yc2EAAAADAQABAAQCoer9ieZTjQ3pXCHczuAYZFj1F7t
+uBkXuqeGMRex78pCvmS+DiEXAMPLEuJ1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdHFzy3guP
+BKc1WqtxJEXAMPLEsBGqZZ1rIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0X1Ms7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHPp440g36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
0t6y9HwvykEXAMPLEAfbKjbr42+u6+0S1kr4d339q2U1sTDytJhhs8HUe11wTfGRfp",
            "witnessedAt": 1570744377.699,
            "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
            "fingerprintSHA256": "SHA256:IEEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
        },
        {
            "algorithm": "ssh-ed25519",
            "publicKey":
"AEXAMPLEaC11ZDI1NTE5AAAAIC1gwGPDfGa0NxEXAMPLEJX3UNap781QxHQmn8nzlrUv",
            "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"
    }
  ]
}
}

```

- Para API obter detalhes, consulte [GetInstanceAccessDetails](#) na Referência de AWS CLI Comandos.

get-instance-metric-data

O código de exemplo a seguir mostra como usar `get-instance-metric-data`.

AWS CLI

Para obter dados métricos de uma instância

O `get-instance-metric-data` exemplo a seguir retorna a porcentagem média de `CPUUtilization` cada 7200 segundo (2 horas) entre 1571342400 e, 1571428800 por exemplo `MEAN-1`.

Recomendamos que você use um conversor de horário unix para identificar os horários de início e término.

```

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

```

Saída:

```
{
  "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"
}
]
```

- Para API obter detalhes, consulte [GetInstanceMetricData](#) na Referência de AWS CLI Comandos.

get-instance-port-states

O código de exemplo a seguir mostra como usar `get-instance-port-states`.

AWS CLI

Para obter informações de firewall para uma instância

O `get-instance-port-states` exemplo a seguir retorna as portas de firewall configuradas, por exemplo `MEAN-1`.

```
aws lightsail get-instance-port-states \
  --instance-name MEAN-1
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetInstancePortStates](#) na Referência de AWS CLI Comandos.

get-instance-snapshot

O código de exemplo a seguir mostra como usar `get-instance-snapshot`.

AWS CLI

Para obter informações sobre um instantâneo de instância especificado

O `get-instance-snapshot` exemplo a seguir exibe detalhes sobre o instantâneo da instância especificada.

```
aws lightsail get-instance-snapshot \
  --instance-snapshot-name MEAN-1-1571419854
```

Saída:

```
{
  "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_4_0_9",
    "fromBundleId": "medium_2_0",
    "isFromAutoSnapshot": false,
    "sizeInGb": 80
  }
}
```

- Para API obter detalhes, consulte [GetInstanceSnapshot](#) na Referência de AWS CLI Comandos.

get-instance-snapshots

O código de exemplo a seguir mostra como usar `get-instance-snapshots`.

AWS CLI

Para obter informações sobre todos os instantâneos da sua instância

O `get-instance-snapshots` exemplo a seguir exibe detalhes sobre todos os instantâneos da instância na AWS região configurada.

```
aws lightsail get-instance-snapshots
```

Saída:


```
{
  "instanceSnapshots": [
    {
      "name": "MEAN-1-1571421498",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
a20e6ebe-b0ee-4ae4-a750-3EXAMPLEcb0c",
      "supportCode": "6EXAMPLE3362/ami-0EXAMPLEEe33cabfa1",
      "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_5_1_1_2",
      "fromBundleId": "micro_2_0",
      "isFromAutoSnapshot": false,
      "sizeInGb": 40
    },
    {
      "name": "MEAN-1-1571419854",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEeac8f",
      "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_4_0_9",
        "fromBundleId": "medium_2_0",
        "isFromAutoSnapshot": false,
        "sizeInGb": 80
    }
]
}
```

- Para API obter detalhes, consulte [GetInstanceSnapshots](#) na Referência de AWS CLI Comandos.

get-instance-state

O código de exemplo a seguir mostra como usar `get-instance-state`.

AWS CLI

Para obter informações sobre o estado de uma instância

O `get-instance-state` exemplo a seguir retorna o estado da instância especificada.

```
aws lightsail get-instance-state \
  --instance-name MEAN-1
```

Saída:

```
{
  "state": {
    "code": 16,
    "name": "running"
  }
}
```

- Para API obter detalhes, consulte [GetInstanceState](#) na Referência de AWS CLI Comandos.

get-instance

O código de exemplo a seguir mostra como usar `get-instance`.

AWS CLI

Para obter informações sobre uma instância

O `get-instance` exemplo a seguir mostra detalhes sobre a instância `MEAN-1`.

```
aws lightsail get-instance \  
  --instance-name MEAN-1
```

Saída:

```
{  
  "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_4_0_9",  
    "blueprintName": "MEAN",  
    "bundleId": "medium_2_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/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": "MyKey"
}
```

```
}
```

- Para API obter detalhes, consulte [GetInstance](#) na Referência de AWS CLI Comandos.

get-instances

O código de exemplo a seguir mostra como usar `get-instances`.

AWS CLI

Para obter informações sobre todas as instâncias

O `get-instances` exemplo a seguir exibe detalhes sobre todas as instâncias na AWS região configurada.

```
aws lightsail get-instances
```

Saída:

```
{
  "instances": [
    {
      "name": "Windows_Server_2016-1",
      "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_2016",
      "blueprintName": "Windows Server 2016",
      "bundleId": "small_win_2_0",
      "isStaticIp": false,
      "privateIpAddress": "192.0.2.0",
      "publicIpAddress": "192.0.2.0",
      "hardware": {
        "cpuCount": 1,
        "disks": [
```

```
    {
      "createdAt": 1571332358.665,
      "sizeInGb": 60,
      "isSystemDisk": true,
      "iops": 180,
      "path": "/dev/sda1",
      "attachedTo": "Windows_Server_2016-1",
      "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_2016-1",
      "isAttached": true,
      "attachmentState": "attached"
    }
  ],
  "ramSizeInGb": 2.0
},
"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_4_0_9",
    "blueprintName": "MEAN",

```

```
"bundleId": "medium_2_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": 100,
      "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"
  }
]
```

- Para API obter detalhes, consulte [GetInstance](#) na Referência de AWS CLI Comandos.

get-key-pair

O código de exemplo a seguir mostra como usar `get-key-pair`.

AWS CLI

Para obter informações sobre um par de chaves

O `get-key-pair` exemplo a seguir exibe detalhes sobre o par de chaves especificado.

```
aws lightsail get-key-pair \  
  --key-pair-name MyKey1
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [GetKeyPair](#) na Referência de AWS CLI Comandos.

get-key-pairs

O código de exemplo a seguir mostra como usar `get-key-pairs`.

AWS CLI

Para obter informações sobre todos os pares de chaves

O `get-key-pairs` exemplo a seguir exibe detalhes sobre todos os pares de chaves na AWS região configurada.

```
aws lightsail get-key-pairs
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetKeyPairs](#) na Referência de AWS CLI Comandos.

`get-load-balancer-tls-certificates`

O código de exemplo a seguir mostra como usar `get-load-balancer-tls-certificates`.

AWS CLI

Para obter informações sobre TLS certificados para um balanceador de carga

O `get-load-balancer-tls-certificates` exemplo a seguir exibe detalhes sobre os TLS certificados do balanceador de carga especificado.

```
aws lightsail get-load-balancer-tls-certificates \
  --load-balancer-name LoadBalancer-1
```

Saída:

```
{
  "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"
        ]
    }
]
}

```

- Para API obter detalhes, consulte [GetLoadBalancerTlsCertificates](#) na Referência de AWS CLI Comandos.

get-load-balancer

O código de exemplo a seguir mostra como usar `get-load-balancer`.

AWS CLI

Para obter informações sobre um balanceador de carga

O `get-load-balancer` exemplo a seguir exibe detalhes sobre o balanceador de carga especificado.

```

aws lightsail get-load-balancer \
  --load-balancer-name LoadBalancer-1

```

Saída:

```

{
  "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"
    }
  }
}
```

- Para API obter detalhes, consulte [GetLoadBalancer](#) na Referência de AWS CLI Comandos.

get-load-balancers

O código de exemplo a seguir mostra como usar `get-load-balancers`.

AWS CLI

Para obter informações sobre todos os balanceadores de carga

O `get-load-balancers` exemplo a seguir exibe detalhes sobre todos os balanceadores de carga na AWS região configurada.

```
aws lightsail get-load-balancers
```

Saída:

```
{
  "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"
}
}
]
}

```

- Para API obter detalhes, consulte [GetLoadBalancers](#) na Referência de AWS CLI Comandos.

get-operation

O código de exemplo a seguir mostra como usar `get-operation`.

AWS CLI

Para obter informações sobre uma única operação

O `get-operation` exemplo a seguir exibe detalhes sobre a operação especificada.

```
aws lightsail get-operation \
  --operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [GetOperation](#) na Referência de AWS CLI Comandos.

get-operations-for-resource

O código de exemplo a seguir mostra como usar `get-operations-for-resource`.

AWS CLI

Para obter todas as operações de um recurso

O `get-operations-for-resource` exemplo a seguir exibe detalhes sobre todas as operações do recurso especificado.

```
aws lightsail get-operations-for-resource \
  --resource-name LoadBalancer-1
```

Saída:

```
{
  "operations": [
    {
      "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
      "resourceName": "LoadBalancer-1",
      "resourceType": "LoadBalancer",
      "createdAt": 1571678786.071,
      "location": {
        "availabilityZone": "all",
        "regionName": "us-west-2"
      }
    },
  ],
}
```

```

    "isTerminal": true,
    "operationDetails": "MEAN-1",
    "operationType": "DetachInstancesFromLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571679087.57
  },
  {
    "id": "2d742a18-0e7f-48c8-9705-3EXAMPLEf98a",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571678782.784,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-1",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571678798.465
  },
  {
    "id": "6c700fcc-4246-40ab-952b-1EXAMPLEd2",
    "resourceName": "LoadBalancer-1",
    "resourceType": "LoadBalancer",
    "createdAt": 1571678775.297,
    "location": {
      "availabilityZone": "all",
      "regionName": "us-west-2"
    },
    "isTerminal": true,
    "operationDetails": "MEAN-3",
    "operationType": "AttachInstancesToLoadBalancer",
    "status": "Succeeded",
    "statusChangedAt": 1571678842.806
  },
  ...
}
]
}

```

- Para API obter detalhes, consulte [GetOperationsForResource](#) na Referência de AWS CLI Comandos.

get-operations

O código de exemplo a seguir mostra como usar `get-operations`.

AWS CLI

Para obter informações sobre todas as operações

O `get-operations` exemplo a seguir exibe detalhes sobre todas as operações na AWS região configurada.

```
aws lightsail get-operations
```

Saída:

```
{
  "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
  },
  ...
}
]
```

- Para API obter detalhes, consulte [GetOperations](#) na Referência de AWS CLI Comandos.

get-regions

O código de exemplo a seguir mostra como usar `get-regions`.

AWS CLI

Para obter todas as AWS regiões do Amazon Lightsail

O `get-regions` exemplo a seguir mostra detalhes sobre todas as AWS regiões do Amazon Lightsail.

```
aws lightsail get-regions
```

Saída:

```
{
```

```

"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": []
  },
  ...
}
]
}

```

- Para API obter detalhes, consulte [GetRegions](#) na Referência de AWS CLI Comandos.

get-relational-database-blueprints

O código de exemplo a seguir mostra como usar `get-relational-database-blueprints`.

AWS CLI

Para obter os planos para novos bancos de dados relacionais

O `get-relational-database-blueprints` exemplo a seguir mostra detalhes sobre todos os esquemas de banco de dados relacional disponíveis que podem ser usados para criar novos bancos de dados relacionais no Amazon Lightsail.

```
aws lightsail get-relational-database-blueprints
```

Saída:

```
{
  "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
  }
]
```

- Para API obter detalhes, consulte [GetRelationalDatabaseBlueprints](#) na Referência de AWS CLI Comandos.

get-relational-database-bundles

O código de exemplo a seguir mostra como usar `get-relational-database-bundles`.

AWS CLI

Para obter os pacotes para novos bancos de dados relacionais

O `get-relational-database-bundles` exemplo a seguir mostra detalhes sobre todos os pacotes de bancos de dados relacionais disponíveis que podem ser usados para criar novos bancos de dados relacionais no Amazon Lightsail. Observe que a resposta não inclui pacotes inativos porque o `--include-inactive` sinalizador não está especificado no comando. Você não pode usar pacotes inativos para criar novos bancos de dados relacionais.

```
aws lightsail get-relational-database-bundles
```

Saída:

```
{
  "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
  }
]
```

```
}
```

Para obter mais informações, consulte [Criação de um banco de dados no Amazon Lightsail](#) no Guia do desenvolvedor do Amazon Lightsail.

- Para API obter detalhes, consulte [GetRelationalDatabaseBundles](#) na Referência de AWS CLI Comandos.

get-relational-database-events

O código de exemplo a seguir mostra como usar `get-relational-database-events`.

AWS CLI

Para obter os eventos para um banco de dados relacional

O `get-relational-database-events` exemplo a seguir exibe detalhes sobre eventos nas últimas 17 horas (1020 minutos) para o banco de dados relacional especificado.

```
aws lightsail get-relational-database-events \
  --relational-database-name Database-1 \
  --duration-in-minutes 1020
```

Saída:

```
{
  "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"
      ]
    }
  ]
}
```

```

    }
  ]
}

```

- Para API obter detalhes, consulte [GetRelationalDatabaseEvents](#) na Referência de AWS CLI Comandos.

get-relational-database-log-events

O código de exemplo a seguir mostra como usar `get-relational-database-log-events`.

AWS CLI

Para obter eventos de log para um banco de dados relacional

O `get-relational-database-log-events` exemplo a seguir exibe detalhes sobre o log especificado entre 1570733176 e 1571597176 para o banco Database1 de dados relacional. As informações retornadas estão configuradas para começar de head.

Recomendamos que você use um conversor de horário unix para identificar os horários de início e término.

```

aws lightsail get-relational-database-log-events \
  --relational-database-name Database1 \
  --log-stream-name error \
  --start-from-head \
  --start-time 1570733176 \
  --end-time 1571597176

```

Saída:

```

{
  "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"
    }
  ]
}

```


get-relational-database-log-streams

O código de exemplo a seguir mostra como usar `get-relational-database-log-streams`.

AWS CLI

Para obter os fluxos de log para um banco de dados relacional

O `get-relational-database-log-streams` exemplo a seguir retorna todos os fluxos de log disponíveis para o banco de dados relacional especificado.

```
aws lightsail get-relational-database-log-streams \
--relational-database-name Database1
```

Saída:

```
{
  "logStreams": [
    "audit",
    "error",
    "general",
    "slowquery"
  ]
}
```

- Para API obter detalhes, consulte [GetRelationalDatabaseLogStreams](#) na Referência de AWS CLI Comandos.

get-relational-database-master-user-password

O código de exemplo a seguir mostra como usar `get-relational-database-master-user-password`.

AWS CLI

Para obter a senha do usuário mestre para um banco de dados relacional

O `get-relational-database-master-user-password` exemplo a seguir retorna informações sobre a senha do usuário mestre para o banco de dados relacional especificado.

```
aws lightsail get-relational-database-master-user-password \
```

```
--relational-database-name Database-1
```

Saída:

```
{
  "masterUserPassword": "VEXAMPLEec.9qvx,_t<)Wkf)kwboM,>2",
  "createdAt": 1571259453.959
}
```

- Para API obter detalhes, consulte [GetRelationalDatabaseMasterUserPassword](#) na Referência de AWS CLI Comandos.

get-relational-database-metric-data

O código de exemplo a seguir mostra como usar `get-relational-database-metric-data`.

AWS CLI

Para obter dados métricos para um banco de dados relacional

O `get-relational-database-metric-data` exemplo a seguir retorna a soma da contagem da métrica `DatabaseConnections` durante o período de 24 horas (86400segundos) entre 1570733176 e 1571597176 para o banco `Database1` de dados relacional.

Recomendamos que você use um conversor de horário unix para identificar os horários de início e término.

```
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
```

Saída:

```
{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [GetRelationalDatabaseMetricData](#) na Referência de AWS CLI Comandos.

get-relational-database-parameters

O código de exemplo a seguir mostra como usar `get-relational-database-parameters`.

AWS CLI

Para obter parâmetros para um banco de dados relacional

O `get-relational-database-parameters` exemplo a seguir retorna informações sobre todos os parâmetros disponíveis para o banco de dados relacional especificado.

```

aws lightsail get-relational-database-parameters \
  --relational-database-name Database-1

```

Saída:

```

{
  "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"
    },
    ...
}
]
}

```

Para obter mais informações, consulte [Atualização de parâmetros de banco de dados no Amazon Lightsail](#) no Lightsail Dev Guide.

- Para API obter detalhes, consulte [GetRelationalDatabaseParameters](#) na Referência de AWS CLI Comandos.

get-relational-database-snapshot

O código de exemplo a seguir mostra como usar `get-relational-database-snapshot`.

AWS CLI

Para obter informações sobre um instantâneo de banco de dados relacional

O `get-relational-database-snapshot` exemplo a seguir exibe detalhes sobre o instantâneo do banco de dados relacional especificado.

```
aws lightsail get-relational-database-snapshot \  
  --relational-database-snapshot-name Database-1-1571350042
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [GetRelationalDatabaseSnapshot](#) na Referência de AWS CLI Comandos.

get-relational-database-snapshots

O código de exemplo a seguir mostra como usar `get-relational-database-snapshots`.

AWS CLI

Para obter informações sobre todos os instantâneos de bancos de dados relacionais

O `get-relational-database-snapshots` exemplo a seguir exibe detalhes sobre todos os instantâneos do banco de dados relacional na região AWS configurada.

```
aws lightsail get-relational-database-snapshots
```

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [GetRelationalDatabaseSnapshots](#) na Referência de AWS CLI Comandos.

get-relational-database

O código de exemplo a seguir mostra como usar `get-relational-database`.

AWS CLI

Para obter informações sobre um banco de dados relacional

O `get-relational-database` exemplo a seguir exibe detalhes sobre o banco de dados relacional especificado.

```
aws lightsail get-relational-database \  
--relational-database-name Database-1
```

Saída:

```
{  
  "relationalDatabase": {  
    "name": "Database-1",  
    "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",  
    "supportCode": "6EXAMPLE3362/1s-9EXAMPLE8ad863723b62cc8901a8aa6e794ae0d2",  
    "createdAt": 1571259453.795,  
    "location": {  
      "availabilityZone": "us-west-2a",  
      "regionName": "us-west-2"  
    },  
    "resourceType": "RelationalDatabase",  
    "tags": [],  
    "relationalDatabaseBlueprintId": "mysql_8_0",  
    "relationalDatabaseBundleId": "micro_1_0",  
    "masterDatabaseName": "dbmaster",  
    "hardware": {  
      "cpuCount": 1,  
      "diskSizeInGb": 40,  
      "ramSizeInGb": 1.0  
    },  
    "state": "available",  
    "backupRetentionEnabled": false,  
    "pendingModifiedValues": {},  
    "engine": "mysql",  
    "engineVersion": "8.0.16",  
    "masterUsername": "dbmasteruser",  
    "parameterApplyStatus": "in-sync",  
    "preferredBackupWindow": "10:01-10:31",  
    "preferredMaintenanceWindow": "sat:11:14-sat:11:44",  
    "publiclyAccessible": true,  
    "masterEndpoint": {  
      "port": 3306,  
      "address": "1s-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-west-2.rds.amazonaws.com"  
    },  
    "pendingMaintenanceActions": []  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [GetRelationalDatabase](#) na Referência de AWS CLI Comandos.

get-relational-databases

O código de exemplo a seguir mostra como usar `get-relational-databases`.

AWS CLI

Para obter informações sobre todos os bancos de dados relacionais

O `get-relational-databases` exemplo a seguir exibe detalhes sobre todos os bancos de dados relacionais na AWS região configurada.

```
aws lightsail get-relational-databases
```

Saída:

```
{
  "relationalDatabases": [
    {
      "name": "MySQL",
      "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",
      "supportCode": "6EXAMPLE3362/1s-3EXAMPLEa995d8c3b06b4501356e5f2f28e1aeba",
      "createdAt": 1554306019.155,
      "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
      },
      "resourceType": "RelationalDatabase",
      "tags": [],
      "relationalDatabaseBlueprintId": "mysql_8_0",
      "relationalDatabaseBundleId": "micro_1_0",
      "masterDatabaseName": "dbmaster",
      "hardware": {
        "cpuCount": 1,
        "diskSizeInGb": 40,
        "ramSizeInGb": 1.0
      }
    }
  ]
}
```

```

    },
    "state": "available",
    "backupRetentionEnabled": true,
    "pendingModifiedValues": {},
    "engine": "mysql",
    "engineVersion": "8.0.15",
    "latestRestorableTime": 1571686200.0,
    "masterUsername": "dbmasteruser",
    "parameterApplyStatus": "in-sync",
    "preferredBackupWindow": "07:51-08:21",
    "preferredMaintenanceWindow": "tue:12:18-tue:12:48",
    "publiclyAccessible": true,
    "masterEndpoint": {
        "port": 3306,
        "address":
"ls-3EXAMPLEa995d8c3b06b4501356e5f2fEXAMPLEa.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
},
{
    "name": "Postgres",
    "arn": "arn:aws:lightsail:us-west-2:111122223333:RelationalDatabase/
e9780b6b-d0ab-4af2-85f1-1EXAMPLEac68",
    "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEb4ffffb5cec056220c734713e14bd5fcd",
    "createdAt": 1554306000.814,
    "location": {
        "availabilityZone": "us-west-2a",
        "regionName": "us-west-2"
    },
    "resourceType": "RelationalDatabase",
    "tags": [],
    "relationalDatabaseBlueprintId": "postgres_11",
    "relationalDatabaseBundleId": "micro_1_0",
    "masterDatabaseName": "dbmaster",
    "hardware": {
        "cpuCount": 1,
        "diskSizeInGb": 40,
        "ramSizeInGb": 1.0
    },
    "state": "available",
    "backupRetentionEnabled": true,
    "pendingModifiedValues": {},

```

```

    "engine": "postgres",
    "engineVersion": "11.1",
    "latestRestorableTime": 1571686339.0,
    "masterUsername": "dbmasteruser",
    "parameterApplyStatus": "in-sync",
    "preferredBackupWindow": "06:19-06:49",
    "preferredMaintenanceWindow": "sun:10:19-sun:10:49",
    "publiclyAccessible": false,
    "masterEndpoint": {
      "port": 5432,
      "address":
        "ls-3EXAMPLEb4ffffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
  }
]
}

```

- Para API obter detalhes, consulte [GetRelationalDatabases](#) na Referência de AWS CLI Comandos.

get-static-ip

O código de exemplo a seguir mostra como usar `get-static-ip`.

AWS CLI

Para obter informações sobre um IP estático

O `get-static-ip` exemplo a seguir exibe detalhes sobre o IP estático especificado.

```

aws lightsail get-static-ip \
  --static-ip-name StaticIp-1

```

Saída:

```

{
  "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
  }
}
```

- Para API obter detalhes, consulte [GetStaticIp](#) na Referência de AWS CLI Comandos.

get-static-ips

O código de exemplo a seguir mostra como usar `get-static-ips`.

AWS CLI

Para obter informações sobre todas as estáticas IPs

O `get-static-ips` exemplo a seguir exibe detalhes sobre toda a estática IPs na AWS região configurada.

```
aws lightsail get-static-ips
```

Saída:

```
{
  "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
    }
]
}

```

- Para API obter detalhes, consulte [GetStaticIps](#) na Referência de AWS CLI Comandos.

is-vpc-peered

O código de exemplo a seguir mostra como usar `is-vpc-peered`.

AWS CLI

Para identificar se sua nuvem privada virtual Amazon Lightsail está emparelhada

O `is-vpc-peered` exemplo a seguir retorna o status de emparelhamento da VPC nuvem privada virtual Amazon Lightsail () para a região especificada. AWS

```
aws lightsail is-vpc-peered \
  --region us-west-2
```

Saída:

```
{
  "isPeered": true
}
```

```
}
```

- Para API obter detalhes, consulte [IsVpcPeered](#) na Referência de AWS CLI Comandos.

open-instance-public-ports

O código de exemplo a seguir mostra como usar `open-instance-public-ports`.

AWS CLI

Para abrir portas de firewall para uma instância

O `open-instance-public-ports` exemplo a seguir abre a TCP porta 22 na instância especificada.

```
aws lightsail open-instance-public-ports \  
  --instance-name MEAN-2 \  
  --port-info fromPort=22,protocol=TCP,toPort=22
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [OpenInstancePublicPorts](#) na Referência de AWS CLI Comandos.

peer-vpc

O código de exemplo a seguir mostra como usar `peer-vpc`.

AWS CLI

Para emparelhar a nuvem privada virtual Amazon Lightsail

O `peer-vpc` exemplo a seguir emparelha a VPC nuvem privada virtual Amazon Lightsail () para a região especificada. AWS

```
aws lightsail peer-vpc \  
  --region us-west-2
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [PeerVpc](#) na Referência de AWS CLI Comandos.

reboot-instance

O código de exemplo a seguir mostra como usar `reboot-instance`.

AWS CLI

Para reinicializar uma instância

O `reboot-instance` exemplo a seguir reinicia a instância especificada.

```
aws lightsail reboot-instance \  
  --instance-name MEAN-1
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [RebootInstance](#) na Referência de AWS CLI Comandos.

reboot-relational-database

O código de exemplo a seguir mostra como usar `reboot-relational-database`.

AWS CLI

Para reinicializar um banco de dados relacional

O `reboot-relational-database` exemplo a seguir reinicia o banco de dados relacional especificado.

```
aws lightsail reboot-relational-database \  
  --instance-name MEAN-1
```

```
--relational-database-name Database-1
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [RebootRelationalDatabase](#) na Referência de AWS CLI Comandos.

release-static-ip

O código de exemplo a seguir mostra como usar `release-static-ip`.

AWS CLI

Para excluir um IP estático

O `release-static-ip` exemplo a seguir exclui o IP estático especificado.

```
aws lightsail release-static-ip \  
  --static-ip-name StaticIp-1
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ReleaseStaticIp](#) na Referência de AWS CLI Comandos.

start-instance

O código de exemplo a seguir mostra como usar `start-instance`.

AWS CLI

Para iniciar uma instância

O `start-instance` exemplo a seguir inicia a instância especificada.

```
aws lightsail start-instance \
  --instance-name WordPress-1
```

Saída:

```
{
  "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
    }
]
}

```

- Para API obter detalhes, consulte [StartInstance](#) na Referência de AWS CLI Comandos.

start-relational-database

O código de exemplo a seguir mostra como usar `start-relational-database`.

AWS CLI

Para iniciar um banco de dados relacional

O `start-relational-database` exemplo a seguir inicia o banco de dados relacional especificado.

```

aws lightsail start-relational-database \
  --relational-database-name Database-1

```

Saída:

```

{
  "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
      }
    ]
  }
```

- Para API obter detalhes, consulte [StartRelationalDatabase](#) na Referência de AWS CLI Comandos.

stop-instance

O código de exemplo a seguir mostra como usar stop-instance.

AWS CLI

Para parar uma instância

O stop-instance exemplo a seguir interrompe a instância especificada.

```
aws lightsail stop-instance \
--instance-name WordPress-1
```

Saída:

```
{
  "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
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [StopInstance](#) na Referência de AWS CLI Comandos.

stop-relational-database

O código de exemplo a seguir mostra como usar `stop-relational-database`.

AWS CLI

Para parar um banco de dados relacional

O `stop-relational-database` exemplo a seguir interrompe o banco de dados relacional especificado.

```
aws lightsail stop-relational-database \
  --relational-database-name Database-1
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [StopRelationalDatabase](#) na Referência de AWS CLI Comandos.

unpeer-vpc

O código de exemplo a seguir mostra como usar `unpeer-vpc`.

AWS CLI

Para unir a nuvem privada virtual Amazon Lightsail

O `unpeer-vpc` exemplo a seguir mostra a nuvem privada virtual Amazon Lightsail VPC () para a região especificada. AWS

```
aws lightsail unpeer-vpc \  
  --region us-west-2
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [UnpeerVpc](#) na Referência de AWS CLI Comandos.

Exemplos de Macie usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Macie.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-buckets

O código de exemplo a seguir mostra como usar `describe-buckets`.

AWS CLI

Para consultar dados sobre um ou mais buckets do S3 que o Amazon Macie monitora e analisa para sua conta

O `describe-buckets` exemplo a seguir consulta os metadados de todos os buckets do S3 cujos nomes começam com MY-S3 e estão na região atual. AWS

```
aws macie2 describe-buckets \  
  --criteria '{"bucketName":{"prefix":"my-S3"}}'
```

Saída:

```
{  
  "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
  }
]
}
```

Para obter mais informações, consulte [Filtrando seu inventário de buckets do S3 no Guia do usuário do Amazon Macie](#).

- Para API obter detalhes, consulte [DescribeBuckets](#) na Referência de AWS CLI Comandos.

Exemplos de Amazon Managed Grafana usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface Amazon Managed Grafana.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

list-workspaces

O código de exemplo a seguir mostra como usar `list-workspaces`.

AWS CLI

Para listar os espaços de trabalho da conta na região especificada pela credencial do usuário

O `list-workspaces` exemplo a seguir lista os espaços de trabalho da Grafana para a região da conta.

```
aws grafana list-workspaces
```

Saída:

```
{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [ListWorkspaces](#) na Referência de AWS CLI Comandos.

MediaConnect exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaConnect.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-flow-outputs

O código de exemplo a seguir mostra como usar add-flow-outputs.

AWS CLI

Para adicionar saídas a um fluxo

O add-flow-outputs exemplo a seguir adiciona saídas ao fluxo especificado.

```
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
```

Saída:

```
{  
  "Outputs": [  
    {  
      "Port": 3333,  
      "OutputArn": "arn:aws:mediaconnect: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"
}

```

Para obter mais informações, consulte [Adicionar saídas a um fluxo no Guia](#) do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [AddFlowOutputs](#) na Referência de AWS CLI Comandos.

create-flow

O código de exemplo a seguir mostra como usar create-flow.

AWS CLI

Para criar um fluxo

O create-flow exemplo a seguir cria um fluxo com a configuração especificada.

```

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

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Criando um fluxo](#) no Guia do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateFlow](#) na Referência de AWS CLI Comandos.

delete-flow

O código de exemplo a seguir mostra como usar delete-flow.

AWS CLI

Para excluir um fluxo

O delete-flow exemplo a seguir exclui o fluxo especificado.

```

aws mediacconnect delete-flow \
  --flow-arn arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow

```

Saída:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Status": "DELETING"
}
```

Para obter mais informações, consulte [Excluindo um fluxo no Guia](#) do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeleteFlow](#) na Referência de AWS CLI Comandos.

describe-flow

O código de exemplo a seguir mostra como usar `describe-flow`.

AWS CLI

Para ver os detalhes de um fluxo

O `describe-flow` exemplo a seguir exibe os detalhes do fluxo especificado ARN, como zona de disponibilidade, status, origem, direitos e saídas.

```
aws mediaconnect describe-flow \
  --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Saída:

```
{
  "Flow": {
    "EgressIp": "54.201.4.39",
    "AvailabilityZone": "us-west-2c",
    "Status": "ACTIVE",
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Entitlements": [
      {
        "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:123456789012:entitlement:1-AaBb11CcDd22EeFf-34DE5fG12AbC:MyEntitlement",
        "Description": "Assign to this account",
```

```
        "Name": "MyEntitlement",
        "Subscribers": [
            "444455556666"
        ]
    },
],
"Description": "NYC awards show",
"Name": "AwardsShow",
"Outputs": [
    {
        "Port": 2355,
        "Name": "NYC",
        "Transport": {
            "SmoothingLatency": 0,
            "Protocol": "rtp-fec"
        },
        "OutputArn": "arn:aws:mediacconnect:us-
east-1:123456789012:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
        "Destination": "192.0.2.0"
    },
    {
        "Port": 3025,
        "Name": "LA",
        "Transport": {
            "SmoothingLatency": 0,
            "Protocol": "rtp-fec"
        },
        "OutputArn": "arn:aws:mediacconnect:us-
east-1:123456789012:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
        "Destination": "192.0.2.0"
    }
],
"Source": {
    "IngestIp": "54.201.4.39",
    "SourceArn": "arn:aws:mediacconnect:us-
east-1:123456789012:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
    "Transport": {
        "MaxBitrate": 80000000,
        "Protocol": "rtp"
    },
    "IngestPort": 1069,
    "Description": "Saturday night show",
    "Name": "ShowSource",
    "WhitelistCidr": "10.24.34.0/23"
```

```

    }
  }
}

```

Para obter mais informações, consulte [Visualizando os detalhes de um fluxo](#) no Guia do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribeFlow](#) na Referência de AWS CLI Comandos.

grant-flow-entitlements

O código de exemplo a seguir mostra como usar grant-flow-entitlements.

AWS CLI

Para conceder um direito em um fluxo

O grant-flow-entitlements exemplo a seguir concede ao fluxo existente especificado o direito de compartilhar seu conteúdo com outra AWS conta.

```

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

```

Saída:

```

{
  "Entitlements": [
    {
      "Name": "AnyCompany_Entitlement",
      "EntitlementArn": "arn:aws:mediaconnect: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"
}

```

Para obter mais informações, consulte [Concedendo um direito em um fluxo no Guia do usuário do AWS Elemental MediaConnect](#).

- Para API obter detalhes, consulte [GrantFlowEntitlements](#) na Referência de AWS CLI Comandos.

list-entitlements

O código de exemplo a seguir mostra como usar `list-entitlements`.

AWS CLI

Para ver uma lista de direitos

O `list-entitlements` exemplo a seguir exibe uma lista de todos os direitos que foram concedidos à conta.

```
aws mediacconnect list-entitlements
```

Saída:

```
{
  "Entitlements": [
    {
      "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
      "EntitlementName": "MyEntitlement"
    }
  ]
}
```

Para obter mais informações, consulte [ListEntitlements](#) na MediaConnect API Referência AWS Elemental.

- Para API obter detalhes, consulte [ListEntitlements](#) na Referência de AWS CLI Comandos.

list-flows

O código de exemplo a seguir mostra como usar `list-flows`.

AWS CLI

Para ver uma lista de fluxos

O `list-flows` exemplo a seguir exibe uma lista de fluxos.

```
aws mediacconnect list-flows
```

Saída:

```
{
  "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:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
    }
]
}

```

Para obter mais informações, consulte [Visualizando uma lista de fluxos](#) no Guia do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListFlows](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um MediaConnect recurso

O `list-tags-for-resource` exemplo a seguir exibe as chaves e os valores de tag associados ao MediaConnect recurso especificado.

```

aws mediaconnect list-tags-for-resource \
  --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame

```

Saída:

```

{
  "Tags": {
    "region": "west",
    "stage": "prod"
  }
}

```

Para obter mais informações, consulte [ListTagsForResource](#), [TagResource](#), [UntagResource](#) na MediaConnect API Referência AWS Elemental.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

remove-flow-output

O código de exemplo a seguir mostra como usar `remove-flow-output`.

AWS CLI

Para remover uma saída de um fluxo

O `remove-flow-output` exemplo a seguir remove uma saída do fluxo especificado.

```
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-c34de5fG6678h:NYC
```

Saída:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG6678h:NYC"
}
```

Para obter mais informações, consulte [Removendo saídas de um fluxo no Guia](#) do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [RemoveFlowOutput](#) na Referência de AWS CLI Comandos.

revoke-flow-entitlement

O código de exemplo a seguir mostra como usar `revoke-flow-entitlement`.

AWS CLI

Para revogar um direito

O `revoke-flow-entitlement` exemplo a seguir revoga um direito no fluxo especificado.

```
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
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Revogar um direito no Guia do usuário](#) do AWS Elemental MediaConnect .

- Para API obter detalhes, consulte [RevokeFlowEntitlement](#) na Referência de AWS CLI Comandos.

start-flow

O código de exemplo a seguir mostra como usar start-flow.

AWS CLI

Para iniciar um fluxo

O start-flow exemplo a seguir inicia o fluxo especificado.

```
aws mediaconnect start-flow \
  --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Este comando não produz saída. Saída:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Status": "STARTING"
}
```

```
}
```

Para obter mais informações, consulte [Iniciando um fluxo](#) no Guia do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [StartFlow](#) na Referência de AWS CLI Comandos.

stop-flow

O código de exemplo a seguir mostra como usar stop-flow.

AWS CLI

Para interromper um fluxo

O stop-flow exemplo a seguir interrompe o fluxo especificado.

```
aws mediaconnect stop-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Saída:

```
{  
  "Status": "STOPPING",  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"  
}
```

Para obter mais informações, consulte [Interromper um fluxo](#) no Guia do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [StopFlow](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para adicionar tags a um MediaConnect recurso

O `tag-resource` exemplo a seguir adiciona uma tag com um nome e valor de chave ao MediaConnect recurso especificado.

```
aws mediacconnect tag-resource \  
  --resource-arn arn:aws:mediacconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \  
  --tags region=west
```

Este comando não produz saída.

Para obter mais informações, consulte [ListTagsForResource](#), [TagResource](#), [UntagResource](#) na MediaConnect APIReferência AWS Elemental.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um MediaConnect recurso

O `untag-resource` exemplo a seguir remove a tag com o nome da chave especificada e seu valor associado de um MediaConnect recurso.

```
aws mediacconnect untag-resource \  
  --resource-arn arn:aws:mediacconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \  
  --tag-keys region
```

Este comando não produz saída.

Para obter mais informações, consulte [ListTagsForResource](#), [TagResource](#), [UntagResource](#) na MediaConnect APIReferência AWS Elemental.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-flow-entitlement

O código de exemplo a seguir mostra como usar `update-flow-entitlement`.

AWS CLI

Para atualizar um direito

O `update-flow-entitlement` exemplo a seguir atualiza o direito especificado com uma nova descrição e assinante.

```
aws mediacconnect update-flow-entitlement \
  --flow-arn arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --entitlement-arn arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement \
  --description 'For AnyCompany Affiliate' \
  --subscribers 777788889999
```

Saída:

```
{
  "FlowArn": "arn:aws:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "Entitlement": {
    "Name": "AnyCompany_Entitlement",
    "Description": "For AnyCompany Affiliate",
    "EntitlementArn": "arn:aws:mediacconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
    "Encryption": {
      "KeyType": "static-key",
      "Algorithm": "aes128",
      "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
      "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
    },
    "Subscribers": [
      "777788889999"
    ]
  }
}
```

Para obter mais informações, consulte [Atualização de um direito no Guia](#) do usuário do AWS Elemental MediaConnect .

- Para API obter detalhes, consulte [UpdateFlowEntitlement](#) na Referência de AWS CLI Comandos.

update-flow-output

O código de exemplo a seguir mostra como usar `update-flow-output`.

AWS CLI

Para atualizar uma saída em um fluxo

O `update-flow-output` exemplo a seguir atualiza uma saída no fluxo especificado.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Atualizando saídas em um fluxo no Guia](#) do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [UpdateFlowOutput](#) na Referência de AWS CLI Comandos.

update-flow-source

O código de exemplo a seguir mostra como usar `update-flow-source`.

AWS CLI

Para atualizar a origem de um fluxo existente

O `update-flow-source` exemplo a seguir atualiza a origem de um fluxo existente.

```
aws mediacconnect update-flow-source \  
  --flow-arn arn:aws:mediacconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \  
  --source-arn arn:aws:mediacconnect:us-  
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource \  
  --description 'Friday night show' \  
  --ingest-port 3344 \  
  --protocol rtp-fec \  
  --whitelist-cidr 10.24.34.0/23
```

Saída:

```
{  
  "FlowArn": "arn:aws:mediacconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",  
  "Source": {  
    "IngestIp": "34.210.136.56",  
    "WhitelistCidr": "10.24.34.0/23",  
    "Transport": {  
      "Protocol": "rtp-fec"  
    },  
    "IngestPort": 3344,  
    "Name": "ShowSource",  
    "Description": "Friday night show",  
    "SourceArn": "arn:aws:mediacconnect:us-  
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"  
  }  
}
```

Para obter mais informações, consulte [Atualizando a fonte de um fluxo](#) no Guia do MediaConnect usuário do AWS Elemental.

- Para API obter detalhes, consulte [UpdateFlowSource](#) na Referência de AWS CLI Comandos.

MediaConvert exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaConvert.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-job

O código de exemplo a seguir mostra como usar `cancel-job`.

AWS CLI

Para cancelar um trabalho que está em uma fila

O `cancel-job` exemplo a seguir cancela o trabalho com ID1234567891234-abc123. Você não pode cancelar um trabalho que o serviço começou a processar.

```
aws mediaconvert cancel-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --id 1234567891234-abc123
```

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Para obter mais informações, consulte [Trabalhando com MediaConvert tarefas AWS elementares no Guia](#) do MediaConvert usuário do AWS Elemental.

- Para API obter detalhes, consulte [CancelJob](#) na Referência de AWS CLI Comandos.

create-job-template

O código de exemplo a seguir mostra como usar `create-job-template`.

AWS CLI

Criar um modelo de trabalho

O `create-job-template` exemplo a seguir cria um modelo de trabalho com as configurações de transcodificação especificadas no arquivo `job-template.json` que reside no seu 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 você criar seu JSON arquivo de modelo de trabalho usando `get-job-template` e depois modificando o arquivo, remova o `JobTemplate` objeto, mas mantenha o objeto secundário de Configurações dentro dele. Além disso, certifique-se de remover os seguintes pares de valores-chave: `LastUpdated`, `ArnType`, e `CreatedAt`. Você pode especificar a categoria, a descrição, o nome e a fila no JSON arquivo ou na linha de comando.

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Se sua solicitação for bem-sucedida, o serviço retornará a JSON especificação do modelo de trabalho que você criou.

Para obter mais informações, consulte [Working with AWS Elemental MediaConvert Job Templates](#) no AWS Elemental MediaConvert User Guide.

- Para API obter detalhes, consulte [CreateJobTemplate](#) na Referência de AWS CLI Comandos.

create-job

O código de exemplo a seguir mostra como usar `create-job`.

AWS CLI

Como criar um trabalho

O exemplo de `create-job` a seguir cria um trabalho de transcodificação com as configurações especificadas em um arquivo `job.json` presente no sistema que você usou para enviar o comando. Essa especificação de JSON trabalho pode especificar cada configuração individualmente, fazer referência a um modelo de trabalho ou referenciar predefinições de saída.

```
aws mediaconvert create-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --cli-input-json file://~/job.json
```

Você pode usar o MediaConvert console AWS Elemental para gerar a especificação do JSON trabalho escolhendo as configurações do trabalho e, em seguida, escolhendo Mostrar trabalho JSON na parte inferior da seção Trabalho.

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Se sua solicitação for bem-sucedida, o serviço retornará a especificação do JSON trabalho que você enviou com sua solicitação.

Para obter mais informações, consulte [Trabalhando com MediaConvert tarefas AWS elementares no Guia](#) do MediaConvert usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateJob](#) na Referência de AWS CLI Comandos.

create-preset

O código de exemplo a seguir mostra como usar `create-preset`.

AWS CLI

Para criar uma predefinição de saída personalizada

O `create-preset` exemplo a seguir cria uma predefinição de saída personalizada com base nas configurações de saída especificadas no arquivo `preset.json`. Você pode especificar a categoria, a descrição e o nome no JSON arquivo ou na linha de 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 você criar seu JSON arquivo predefinido usando `get-preset` e modificando o arquivo de saída, certifique-se de remover os seguintes pares de valores-chave: `LastUpdated`, e `Arn` `Type` `CreatedAt`

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Para obter mais informações, consulte [Trabalhando com predefinições de MediaConvert saída AWS elementares](#) no Guia do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [CreatePreset](#) na Referência de AWS CLI Comandos.

create-queue

O código de exemplo a seguir mostra como usar `create-queue`.

AWS CLI

Para criar uma fila personalizada

O `create-queue` exemplo a seguir cria uma fila de transcodificação personalizada.

```
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."
```

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Saída:

```
{  
  "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."  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Trabalhando com MediaConvert filas AWS elementares no Guia](#) do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [CreateQueue](#) na Referência de AWS CLI Comandos.

delete-job-template

O código de exemplo a seguir mostra como usar delete-job-template.

AWS CLI

Para excluir um modelo de trabalho

O delete-job-template exemplo a seguir exclui o modelo de trabalho personalizado especificado.

```
aws mediaconvert delete-job-template \  
  --name "DASH Streaming" \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Este comando não produz saída. Execute `aws mediaconvert list-job-templates` para confirmar que seu modelo foi excluído.

Para obter mais informações, consulte [Working with AWS Elemental MediaConvert Job Templates](#) no AWS Elemental MediaConvert User Guide.

- Para API obter detalhes, consulte [DeleteJobTemplate](#) na Referência de AWS CLI Comandos.

delete-preset

O código de exemplo a seguir mostra como usar delete-preset.

AWS CLI

Para excluir uma fila sob demanda personalizada

O delete-preset exemplo a seguir exclui a predefinição personalizada especificada.

```
aws mediaconvert delete-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

```
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Este comando não produz saída. Execute `aws mediaconvert list-presets` para confirmar que sua predefinição foi excluída.

Para obter mais informações, consulte [Trabalhando com predefinições de MediaConvert saída AWS elementares](#) no Guia do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [DeletePreset](#) na Referência de AWS CLI Comandos.

delete-queue

O código de exemplo a seguir mostra como usar `delete-queue`.

AWS CLI

Para excluir uma fila sob demanda personalizada

O `delete-queue` exemplo a seguir exclui a fila sob demanda personalizada especificada.

Você não pode excluir sua fila padrão. Não é possível excluir uma fila reservada que tenha um plano de preços ativo ou que contenha tarefas não processadas.

```
aws mediaconvert delete-queue \  
  --name Customer1 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Este comando não produz saída. Execute `aws mediaconvert list-queues` para confirmar que sua fila foi excluída.

Para obter mais informações, consulte [Trabalhando com MediaConvert filas AWS elementares no Guia](#) do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [DeleteQueue](#) na Referência de AWS CLI Comandos.

describe-endpoints

O código de exemplo a seguir mostra como usar `describe-endpoints`.

AWS CLI

Para obter o endpoint específico da sua conta

O `describe-endpoints` exemplo a seguir recupera o endpoint necessário para enviar qualquer outra solicitação ao serviço.

```
aws mediaconvert describe-endpoints
```

Saída:

```
{
  "Endpoints": [
    {
      "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"
    }
  ]
}
```

Para obter mais informações, consulte [Introdução ao MediaConvert uso do API](#) na MediaConvert API Referência AWS Elemental.

- Para API obter detalhes, consulte [DescribeEndpoints](#) na Referência de AWS CLI Comandos.

get-job-template

O código de exemplo a seguir mostra como usar `get-job-template`.

AWS CLI

Para obter detalhes de um modelo de trabalho

O `get-job-template` exemplo a seguir exibe a JSON definição do modelo de trabalho personalizado especificado.

```
aws mediaconvert get-job-template \
  --name "DASH Streaming" \
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```

Saída:

```
{
  "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"
}
}
```

Para obter mais informações, consulte [Working with AWS Elemental MediaConvert Job Templates](#) no AWS Elemental MediaConvert User Guide.

- Para API obter detalhes, consulte [GetJobTemplate](#) na Referência de AWS CLI Comandos.

get-job

O código de exemplo a seguir mostra como usar `get-job`.

AWS CLI

Como obter detalhes de um trabalho específico

O exemplo a seguir solicita as informações do trabalho com o ID `1234567890987-1ab2c3` que, neste exemplo, resultou em erro.

```
aws mediaconvert get-job \
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
  --region region-name-1 \
  --id 1234567890987-1ab2c3
```

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Se sua solicitação for bem-sucedida, o serviço retornará um JSON arquivo com informações do trabalho, incluindo configurações do trabalho, quaisquer erros retornados e outros dados do trabalho, da seguinte forma:

```
{
```

```

"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
}
}

```

Para obter mais informações, consulte [Trabalhando com MediaConvert tarefas AWS elementares no Guia](#) do MediaConvert usuário do AWS Elemental.

- Para API obter detalhes, consulte [GetJob](#) na Referência de AWS CLI Comandos.

get-preset

O código de exemplo a seguir mostra como usar get-preset.

AWS CLI

Para obter detalhes de uma predefinição específica

O get-preset exemplo a seguir solicita a JSON definição da predefinição personalizada especificada.

```

aws mediaconvert get-preset \
  --name SimpleMP4 \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com

```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Trabalhando com predefinições de MediaConvert saída AWS elementares](#) no Guia do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [GetPreset](#) na Referência de AWS CLI Comandos.

get-queue

O código de exemplo a seguir mostra como usar get-queue.

AWS CLI

Para obter detalhes de uma fila

O get-queue exemplo a seguir recupera os detalhes da fila personalizada especificada.

```
aws mediaconvert get-queue \  
  --name Customer1 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com MediaConvert filas AWS elementares no Guia](#) do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [GetQueue](#) na Referência de AWS CLI Comandos.

list-job-templates

O código de exemplo a seguir mostra como usar `list-job-templates`.

AWS CLI

Exemplo 1: Para listar seus modelos de trabalho personalizados

O `list-job-templates` exemplo a seguir lista todos os modelos de trabalho personalizados na região atual. Para listar os modelos de trabalho do sistema, veja o próximo exemplo.

```
aws mediaconvert list-job-templates \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Saída:

```
{
  "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
      }
    ]
  }

```

Exemplo 2: Para listar os modelos MediaConvert de trabalho do sistema

O `list-job-templates` exemplo a seguir lista todos os modelos de trabalho do sistema.

```

aws mediaconvert list-job-templates \
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \
  --list-by SYSTEM

```

Saída:

```

{
  "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>...
]

```

```
}
```

Para obter mais informações, consulte [Working with AWS Elemental MediaConvert Job Templates](#) no AWS Elemental MediaConvert User Guide.

- Para API obter detalhes, consulte [ListJobTemplates](#) na Referência de AWS CLI Comandos.

list-jobs

O código de exemplo a seguir mostra como usar `list-jobs`.

AWS CLI

Como obter detalhes de todos os trabalhos de uma região

O exemplo a seguir solicita as informações de todos os trabalhos na região especificada.

```
aws mediaconvert list-jobs \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1
```

Use `describe-endpoints` para obter o endpoint específico da sua conta ou envie o comando sem o endpoint. O serviço retorna um erro e o endpoint.

Para obter mais informações, consulte [Trabalhando com MediaConvert tarefas AWS elementares no Guia](#) do MediaConvert usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListJobs](#) na Referência de AWS CLI Comandos.

list-presets

O código de exemplo a seguir mostra como usar `list-presets`.

AWS CLI

Exemplo 1: Para listar suas predefinições de saída personalizadas

O `list-presets` exemplo a seguir lista suas predefinições de saída personalizadas. Para listar as predefinições do sistema, veja o próximo exemplo.

```
aws mediaconvert list-presets \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Saída:

```
{
  "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."
    }
  ]
}
```

Exemplo 2: Para listar as predefinições de saída do sistema

O `list-presets` exemplo a seguir lista as predefinições MediaConvert de sistema disponíveis. Para listar suas predefinições personalizadas, veja o exemplo anterior.

```
aws mediaconvert list-presets \
  --list-by SYSTEM \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Saída:

```
{
  "Presets": [
    {
```

```

    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/System-
Avc_16x9_1080p_29_97fps_8500kbps",
    "Name": "System-Avc_16x9_1080p_29_97fps_8500kbps",
    "CreatedAt": 1568321789,
    "Description": "Wifi, 1920x1080, 16:9, 29.97fps, 8500kbps",
    "LastUpdated": 1568321789,
    "Type": "SYSTEM",
    "Category": "HLS",
    "Settings": {
      ...<output settings removed for brevity>...
    }
  },
  ...<list of presets shortened for brevity>...

  {
    "Arn": "arn:aws:mediaconvert:us-east-1:123456789012:presets/System-
Xdcam_HD_1080i_29_97fps_35mpbs",
    "Name": "System-Xdcam_HD_1080i_29_97fps_35mpbs",
    "CreatedAt": 1568321790,
    "Description": "XDCAM MPEG HD, 1920x1080i, 29.97fps, 35mbps",
    "LastUpdated": 1568321790,
    "Type": "SYSTEM",
    "Category": "MXF",
    "Settings": {
      ...<output settings removed for brevity>...
    }
  }
]
}

```

Para obter mais informações, consulte [Trabalhando com predefinições de MediaConvert saída AWS elementares](#) no Guia do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [ListPresets](#) na Referência de AWS CLI Comandos.

list-queues

O código de exemplo a seguir mostra como usar `list-queues`.

AWS CLI

Para listar suas filas

O `list-queues` exemplo a seguir lista todas as suas MediaConvert filas.

```
aws mediaconvert list-queues \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Saída:

```
{  
  "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
  }
]
}

```

Para obter mais informações, consulte [Trabalhando com MediaConvert filas AWS elementares no Guia](#) do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [ListQueues](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags em uma MediaConvert fila, modelo de trabalho ou predefinição de saída

O `list-tags-for-resource` exemplo a seguir lista as tags na predefinição de saída especificada.

```

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

```

Saída:

```

{
  "ResourceTags": {
    "Tags": {
      "customer": "zippyVideo"
    },
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"
  }
}

```

Para obter mais informações, consulte Como [marcar MediaConvert filas AWS elementares, modelos de trabalho e predefinições de saída](#) no Guia do usuário do AWS Elemental.

MediaConvert

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

update-job-template

O código de exemplo a seguir mostra como usar `update-job-template`.

AWS CLI

Para alterar um modelo de trabalho

O `update-job-template` exemplo a seguir substitui a JSON definição do modelo de trabalho personalizado especificado pela JSON definição no arquivo fornecido.

```
aws mediaconvert update-job-template --name Arquivo1 https://abcd1234.mediaconvert.us-west-2.amazonaws.com --endpoint-url --arquivo: //~/json cli-input-json job-template-update
```

Conteúdo de `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
}

```

O sistema retorna a JSON carga que você envia com sua solicitação, mesmo quando a solicitação resulta em um erro. Portanto, o JSON retornado não é necessariamente a nova definição do modelo de trabalho.

Como a JSON carga útil pode ser longa, talvez seja necessário rolar para cima para ver as mensagens de erro.

Para obter mais informações, consulte [Working with AWS Elemental MediaConvert Job Templates](#) no AWS Elemental MediaConvert User Guide.

- Para API obter detalhes, consulte [UpdateJobTemplate](#) na Referência de AWS CLI Comandos.

update-preset

O código de exemplo a seguir mostra como usar update-preset.

AWS CLI

Para alterar uma predefinição

O update-preset exemplo a seguir substitui a descrição da predefinição especificada.

```
aws mediaconvert update-preset \  
--name Customer1 \  
--description "New description text." \  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Este comando não produz saída. Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com predefinições de MediaConvert saída AWS elementares](#) no Guia do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [UpdatePreset](#) na Referência de AWS CLI Comandos.

update-queue

O código de exemplo a seguir mostra como usar update-queue.

AWS CLI

Para alterar uma fila

O `update-queue` exemplo a seguir pausa a fila especificada, alterando seu status para `PAUSED`

```
aws mediaconvert update-queue \  
--name Customer1 \  
--status PAUSED \  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com MediaConvert filas AWS elementares no Guia](#) do usuário do AWS Elemental MediaConvert .

- Para API obter detalhes, consulte [UpdateQueue](#) na Referência de AWS CLI Comandos.

MediaLive exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaLive.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-channel

O código de exemplo a seguir mostra como usar `create-channel`.

AWS CLI

Para criar um canal

O `create-channel` exemplo a seguir cria um canal passando um JSON arquivo que contém os parâmetros que você deseja especificar.

O canal neste exemplo ingere uma HLS PULL entrada que se conecta a uma fonte que contém vídeo, áudio e legendas incorporadas. O canal cria um grupo HLS de saída com um servidor Akamai como destino. O grupo de saída contém duas saídas: uma para vídeo e AAC áudio H.265 e outra para VTT legendas da Web, somente em inglês.

O canal deste JSON exemplo inclui os parâmetros mínimos necessários para um canal que usa uma HLS PULL entrada e produz um grupo de HLS saída com a Akamai como destino. O JSON contém as seguintes seções principais:

`InputAttachments`, que especifica uma fonte para o áudio e uma fonte para as legendas. Ele não especifica um seletor de vídeo, o que significa que MediaLive extrai o primeiro vídeo encontrado na fonte. `Destinations`, que contém os dois endereços IP (URLs) para o único grupo de saída nesse canal. Esses endereços exigem senhas. `EncoderSettings`, que contém subseções. `AudioDescriptions`, que especifica que o canal contém um ativo de saída de áudio, que usa a fonte de `InputAttachments` e produz áudio em AAC formato. `CaptionDescriptions`, que especifica que o canal contém um ativo de saída de legendas, que usa a fonte de `InputAttachments` e produz legendas no formato Web. VTT `VideoDescriptions`, que especifica que o canal contém um ativo de saída de vídeo, com a resolução especificada. `OutputGroups`, que especifica os grupos de saída. Neste exemplo, há um grupo chamado `Akamai`. A conexão é feita usando `HLSPUT`. O grupo de saída contém duas saídas. Uma saída é para o ativo de vídeo (nomeado `Video_high`) e o ativo de áudio (nomeado `Audio_EN`). Uma saída é para o ativo de legendas (nomeado `WebVTT_EN`).

Neste exemplo, alguns dos parâmetros não contêm valor ou contêm parâmetros vazios aninhados. Por `OutputSettings` exemplo, pois a `Video_and_audio` saída contém vários

parâmetros aninhados que terminam em um parâmetro vazio `M3U8Settings`. Esse parâmetro deve ser incluído, mas você pode omitir um, vários ou todos os seus filhos, o que significa que o filho assumirá seu valor padrão ou será nulo.

Todos os parâmetros que se aplicam a esse canal de exemplo, mas que não estão especificados nesse arquivo, assumirão o valor padrão, serão definidos como null ou terão um valor exclusivo gerado por MediaLive.

```
aws medialive create-channel \  
  --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

Conteúdo de `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"
  }
}
}
```

Saída:

A saída repete o conteúdo do JSON arquivo, mais os valores a seguir. Todos os parâmetros são ordenados alfabeticamente.

ARN para o canal. A última parte do ARN é o ID exclusivo do canal. `EgressEndpoint` está em branco neste canal de exemplo porque é usado somente para PUSH entradas. Quando aplicável, mostra os endereços para os MediaLive quais o conteúdo foi enviado. `OutputGroups`, `Outputs`. Eles mostram todos os parâmetros do grupo de saída e das saídas, incluindo aqueles que você não incluiu, mas que são relevantes para esse canal. Os parâmetros podem estar vazios (talvez indicando que o parâmetro ou recurso está desativado nessa configuração de canal) ou podem mostrar o valor padrão que será aplicado. `LogLevel` está definido como padrão (DISABLED). `Tags` é definido como padrão (nulo). `PipelinesRunningCount` State mostre o status atual do canal.

Para obter mais informações, consulte [Criando um canal do zero](#) no Guia do MediaLive usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateChannel](#) na Referência de AWS CLI Comandos.

create-input

O código de exemplo a seguir mostra como usar `create-input`.

AWS CLI

Para criar uma entrada

O `create-input` exemplo a seguir cria uma HLS PULL entrada passando um JSON arquivo que contém os parâmetros que se aplicam a esse tipo de entrada. A entrada deste JSON exemplo especifica duas fontes (endereços) para a entrada, a fim de oferecer suporte à redundância na ingestão. Esses endereços exigem senhas.


```
aws medialive create-input \  
--cli-input-json file://input-hls-pull-news.json
```

Conteúdo de 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"  
}
```

Saída:

A saída repete o conteúdo do JSON arquivo, mais os valores a seguir. Todos os parâmetros são ordenados alfabeticamente.

`Arn` para a entrada. A última parte do ARN é o ID de entrada exclusivo. `Attached Channels`, que está sempre vazio para uma entrada recém-criada. `Destinations`, que está vazio neste exemplo porque é usado somente com uma PUSH entrada. `Id` para a entrada, o mesmo que o ID no ARN. `MediaConnectFlows`, que está vazio neste exemplo porque é usado somente com uma entrada do tipo MediaConnect. `SecurityGroups`, que está vazio neste exemplo porque é usado somente com uma PUSH entrada. `Statedessa` entrada. `Tags`, que está vazio (o padrão para esse parâmetro).

Para obter mais informações, consulte [Criação de uma entrada](#) no Guia do MediaLive usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateInput](#) na Referência de AWS CLI Comandos.

MediaPackage exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaPackage.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-channel

O código de exemplo a seguir mostra como usar `create-channel`.

AWS CLI

Para criar um canal

O `create-channel` comando a seguir cria um canal chamado `sportschannel` na conta atual.

```
aws mediapackage create-channel --id sportschannel
```

Saída:

```
{
  "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"
}
}

```

Para obter mais informações, consulte [Criação de um canal](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateChannel](#) na Referência de AWS CLI Comandos.

create-origin-endpoint

O código de exemplo a seguir mostra como usar `create-origin-endpoint`.

AWS CLI

Para criar um endpoint de origem

O `create-origin-endpoint` comando a seguir cria um endpoint de origem nomeado `cmf sports` com as configurações de pacote fornecidas em um JSON arquivo e as configurações de endpoint especificadas.

```

aws mediapackage create-origin-endpoint \
  --channel-id sportschannel \
  --id cmf sports \
  --cmf-package file:///file/path/cmafpkg.json --description "cmf output of sports" \
  --id cmf_sports \
  --manifest-name sports_channel \

```

```
--startover-window-seconds 300 \  
--tags region=west,media=sports \  
--time-delay-seconds 10
```

Saída:

```
{  
  "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": []  
}
```

Para obter mais informações, consulte [Criação de um endpoint no Guia](#) do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateOriginEndpoint](#) na Referência de AWS CLI Comandos.

delete-channel

O código de exemplo a seguir mostra como usar `delete-channel`.

AWS CLI

Para excluir um canal

O `delete-channel` comando a seguir exclui o canal chamado `test`.

```
aws mediapackage delete-channel \  
  --id test
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um canal no Guia](#) do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeleteChannel](#) na Referência de AWS CLI Comandos.

delete-origin-endpoint

O código de exemplo a seguir mostra como usar `delete-origin-endpoint`.

AWS CLI

Para excluir um endpoint de origem

O `delete-origin-endpoint` comando a seguir exclui o endpoint de origem chamado `tester2`

```
aws mediapackage delete-origin-endpoint \  
  --id tester2
```

Para obter mais informações, consulte [Excluindo um endpoint](#) no Guia do usuário do AWS Elemental MediaPackage .

- Para API obter detalhes, consulte [DeleteOriginEndpoint](#) na Referência de AWS CLI Comandos.

describe-channel

O código de exemplo a seguir mostra como usar `describe-channel`.

AWS CLI

Para descrever um canal

O `describe-channel` comando a seguir exibe todos os detalhes do canal nomeado `test`.

```
aws mediapackage describe-channel \  
  --id test
```

Saída:

```
{  
  "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": {}  
}
```

Para obter mais informações, consulte Visualizando detalhes do canal < <https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html> > no Guia do usuário do Elemental AWS MediaPackage

- Para API obter detalhes, consulte [DescribeChannel](#) na Referência de AWS CLI Comandos.

describe-origin-endpoint

O código de exemplo a seguir mostra como usar `describe-origin-endpoint`.

AWS CLI

Para descrever um endpoint de origem

O `describe-origin-endpoint` comando a seguir exibe todos os detalhes do endpoint de origem chamado `cmaf_sports`.

```
aws mediapackage describe-origin-endpoint \  
  --id cmaf_sports
```

Saída:

```
{  
  "Arn": "arn:aws:mediapackage:us-  
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "NONE",  
        "Id": "cmaf_sports_endpoint",  
        "IncludeIframeOnlyStream": false,  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 60,  
        "ProgramDateTimeIntervalSeconds": 0,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-  
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/  
index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  }  
}
```

```
},
  "Id": "cmf_sports",
  "ManifestName": "index",
  "StartoverWindowSeconds": 0,
  "Tags": {
    "region": "west",
    "media": "sports"
  },
  "TimeDelaySeconds": 0,
  "Url": "",
  "Whitelist": []
}
```

Para obter mais informações, consulte [Visualizando um único endpoint no Guia do MediaPackage usuário do AWS Elemental](#).

- Para API obter detalhes, consulte [DescribeOriginEndpoint](#) na Referência de AWS CLI Comandos.

list-channels

O código de exemplo a seguir mostra como usar `list-channels`.

AWS CLI

Como listar todos os canais

O `list-channels` comando a seguir lista todos os canais que estão configurados na AWS conta atual.

```
aws mediapackage list-channels
```

Saída:

```
{
  "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": {}
}
]
}

```

Para obter mais informações, consulte [Visualizando detalhes do canal](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListChannels](#) na Referência de AWS CLI Comandos.

list-origin-endpoints

O código de exemplo a seguir mostra como usar `list-origin-endpoints`.

AWS CLI

Como listar todos os endpoints de origem de um canal

O comando `list-origin-endpoints` a seguir lista todos endpoints de origem que estão configurados no canal `test`.

```
aws mediapackage list-origin-endpoints \
  --channel-id test
```

Saída:

```
{
  "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": []
}
]
}

```

Para obter mais informações, consulte [Visualização de todos os endpoints associados a um canal no Guia](#) do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListOriginEndpoints](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags atribuídas a um recurso

O `list-tags-for-resource` comando a seguir lista as tags atribuídas ao recurso especificado.

```

aws mediapackage list-tags-for-resource \
  --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0

```

Saída:

```
{
```

```
"Tags": {
  "region": "west"
}
```

Para obter mais informações, consulte Como [marcar recursos no AWS Elemental MediaPackage no Guia](#) do usuário do AWS Elemental MediaPackage .

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

rotate-ingest-endpoint-credentials

O código de exemplo a seguir mostra como usar `rotate-ingest-endpoint-credentials`.

AWS CLI

Para alternar as credenciais de ingestão

O `rotate-ingest-endpoint-credentials` comando a seguir alterna o nome de DAV usuário e a senha da Web para o endpoint de ingestão especificado.

```
aws mediapackage rotate-ingest-endpoint-credentials \
  --id test \
  --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

Saída:

```
{
  "Arn": "arn:aws:mediapackage:us-
west-2:111222333:channels/584797f1740548c389a273585dd22a63",
  "HlsIngest": {
    "IngestEndpoints": [
      {
        "Id": "584797f1740548c389a273585dd22a63",
        "Password": "webdavregeneratedpassword1",
        "Url": "https://9be9c4405c474882.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/584797f1740548c389a273585dd22a63/channel",
        "Username": "webdavregeneratedusername1"
      },
      {
```

```

        "Id": "7d187c8616fd455f88aaa5a9fcf74442",
        "Password": "webdavgeneratedpassword2",
        "Url": "https://7bf454c57220328d.mediapackage.us-
west-2.amazonaws.com/in/
v2/584797f1740548c389a273585dd22a63/7d187c8616fd455f88aaa5a9fcf74442/channel",
        "Username": "webdavgeneratedusername2"
    }
]
},
"Id": "test",
"Tags": {}
}

```

Para obter mais informações, consulte [Rotação de credenciais em uma entrada URL no Guia](#) do usuário do AWS Elemental MediaPackage .

- Para API obter detalhes, consulte [RotateIngestEndpointCredentials](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para adicionar uma tag a um recurso

Os tag-resource comandos a seguir adicionam um par de region=west chave e valor ao recurso especificado.

```

aws mediapackage tag-resource \
  --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
  --tags region=west

```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar recursos no AWS Elemental MediaPackage no Guia](#) do usuário do AWS Elemental MediaPackage .

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resource` comando a seguir remove a tag com a chave `region` do canal especificado.

```
aws mediapackage untag-resource \  
  --resource-arn arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \  
  --tag-keys region
```

Para obter mais informações, consulte Como [marcar recursos no AWS Elemental MediaPackage no Guia](#) do usuário do AWS Elemental MediaPackage .

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-channel

O código de exemplo a seguir mostra como usar `update-channel`.

AWS CLI

Para atualizar um canal

O `update-channel` comando a seguir atualiza o canal nomeado `sportschannel` para incluir a descrição `24x7 sports`.

```
aws mediapackage update-channel \  
  --id sportschannel \  
  --description "24x7 sports"
```

Saída:

```
{  
  "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": {}
  }
}

```

Para obter mais informações, consulte [Editando um canal](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [UpdateChannel](#) na Referência de AWS CLI Comandos.

update-origin-endpoint

O código de exemplo a seguir mostra como usar `update-origin-endpoint`.

AWS CLI

Para atualizar um endpoint de origem

O `update-origin-endpoint` comando a seguir atualiza o endpoint de origem chamado `cmf_sports`. Ele altera o tempo de atraso para `0` segundos.

```
aws mediapackage update-origin-endpoint \
```

```
--id cmaf_sports \  
--time-delay-seconds 0
```

Saída:

```
{  
  "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": []  
}
```

Para obter mais informações, consulte [Editando um endpoint no Guia](#) do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [UpdateOriginEndpoint](#) na Referência de AWS CLI Comandos.

MediaPackage VOD exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaPackage VOD.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-asset

O código de exemplo a seguir mostra como usar `create-asset`.

AWS CLI

Para criar um ativo

O `create-asset` exemplo a seguir cria um ativo nomeado `Chicken_Asset` na AWS conta atual. O ativo também ingere o arquivo `30sec_chicken.smil`. 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
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [Ingest an Asset no Guia](#) do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreateAsset](#) na Referência de AWS CLI Comandos.

create-packaging-configuration

O código de exemplo a seguir mostra como usar `create-packaging-configuration`.

AWS CLI

Para criar uma configuração de empacotamento

O `create-packaging-configuration` exemplo a seguir cria uma configuração de empacotamento nomeada `new_hls` no grupo de pacotes chamado `hls_chicken`. Este exemplo usa um arquivo no disco chamado `hls_pc.json` para fornecer os detalhes.

```

aws mediapackage-vod create-packaging-configuration \
  --id new_hls \
  --packaging-group-id hls_chicken \
  --hls-package file://hls_pc.json

```

Conteúdo de `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
}
```

Saída:

```
{
  "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"
        }
      }
    ]
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Criação de uma configuração de empacotamento](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreatePackagingConfiguration](#) na Referência de AWS CLI Comandos.

create-packaging-group

O código de exemplo a seguir mostra como usar `create-packaging-group`.

AWS CLI

Para criar um grupo de embalagens

O `create-packaging-group` exemplo a seguir lista todos os grupos de pacotes que estão configurados na AWS conta atual.

```
aws mediapackage-vod create-packaging-group \  
  --id hls_chicken
```

Saída:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/  
hls_chicken",  
  "Id": "hls_chicken"  
}
```

Para obter mais informações, consulte [Criação de um grupo de pacotes](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [CreatePackagingGroup](#) na Referência de AWS CLI Comandos.

delete-asset

O código de exemplo a seguir mostra como usar `delete-asset`.

AWS CLI

Para excluir um ativo

O `delete-asset` exemplo a seguir exclui o ativo chamado `30sec_chicken`.

```
aws mediapackage-vod delete-asset \  
  --id 30sec_chicken
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um ativo no Guia](#) do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeleteAsset](#) na Referência de AWS CLI Comandos.

`delete-packaging-configuration`

O código de exemplo a seguir mostra como usar `delete-packaging-configuration`.

AWS CLI

Para excluir uma configuração de empacotamento

O `delete-packaging-configuration` exemplo a seguir exclui a configuração de empacotamento chamada `CMAF`.

```
aws mediapackage-vod delete-packaging-configuration \  
  --id CMAF
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo uma configuração de empacotamento](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeletePackagingConfiguration](#) na Referência de AWS CLI Comandos.

`delete-packaging-group`

O código de exemplo a seguir mostra como usar `delete-packaging-group`.

AWS CLI

Para excluir um grupo de embalagens

O `delete-packaging-group` exemplo a seguir exclui o grupo de pacotes chamado `Dash_widevine`.

```
aws mediapackage-vod delete-packaging-group \  
  --id Dash_widevine
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um grupo de pacotes no Guia do MediaPackage](#) usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeletePackagingGroup](#) na Referência de AWS CLI Comandos.

describe-asset

O código de exemplo a seguir mostra como usar `describe-asset`.

AWS CLI

Para descrever um ativo

O `describe-asset` exemplo a seguir exibe todos os detalhes do ativo chamado `30sec_chicken`.

```
aws mediapackage-vod describe-asset \  
  --id 30sec_chicken
```

Saída:

```
{  
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Visualizando detalhes do ativo](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribeAsset](#) na Referência de AWS CLI Comandos.

describe-packaging-configuration

O código de exemplo a seguir mostra como usar `describe-packaging-configuration`.

AWS CLI

Para descrever uma configuração de empacotamento

O `describe-packaging-configuration` exemplo a seguir exibe todos os detalhes da configuração de empacotamento chamada DASH.

```
aws mediapackage-vod describe-packaging-configuration \
  --id DASH
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Exibindo detalhes da configuração do empacotamento](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribePackagingConfiguration](#) na Referência de AWS CLI Comandos.

describe-packaging-group

O código de exemplo a seguir mostra como usar describe-packaging-group.

AWS CLI

Para descrever um grupo de embalagens

O describe-packaging-group exemplo a seguir exibe todos os detalhes do grupo de embalagens chamado Packaging_group_1.

```
aws mediapackage-vod describe-packaging-group \
  --id Packaging_group_1
```

Saída:


```
{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
Packaging_group_1",
  "Id": "Packaging_group_1"
}
```

Para obter mais informações, consulte [Visualizando detalhes do grupo de empacotamento](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribePackagingGroup](#) na Referência de AWS CLI Comandos.

list-assets

O código de exemplo a seguir mostra como usar `list-assets`.

AWS CLI

Para listar todos os ativos

O `list-assets` exemplo a seguir lista todos os ativos que estão configurados na AWS conta atual.

```
aws mediapackage-vod list-assets
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Visualizando detalhes do ativo](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListAssets](#) na Referência de AWS CLI Comandos.

list-packaging-configurations

O código de exemplo a seguir mostra como usar `list-packaging-configurations`.

AWS CLI

Para listar todas as configurações de embalagem

O `list-packaging-configurations` exemplo a seguir lista todas as configurações de empacotamento que estão configuradas no grupo de pacotes chamado `Packaging_group_1`.

```
aws mediapackage-vod list-packaging-configurations \
  --packaging-group-id Packaging_group_1
```

Saída:

```
{
  "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"
        }
      }
    ]
  }
}

```

Para obter mais informações, consulte [Exibindo detalhes da configuração do empacotamento](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListPackagingConfigurations](#) na Referência de AWS CLI Comandos.

list-packaging-groups

O código de exemplo a seguir mostra como usar `list-packaging-groups`.

AWS CLI

Para listar todos os grupos de embalagens

O `list-packaging-groups` exemplo a seguir lista todos os grupos de pacotes que estão configurados na AWS conta atual.

```
aws mediapackage-vod list-packaging-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Visualizando detalhes do grupo de empacotamento](#) no Guia do MediaPackage usuário do AWS Elemental.

- Para API obter detalhes, consulte [ListPackagingGroups](#) na Referência de AWS CLI Comandos.

MediaStore Exemplos de planos de dados usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com MediaStore Data Plane.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-object

O código de exemplo a seguir mostra como usar `delete-object`.

AWS CLI

Para excluir um objeto

O `delete-object` exemplo a seguir exclui o objeto especificado.

```
aws mediastore-data delete-object \  
  --endpoint=https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path=/folder_name/README.md
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um objeto no Guia](#) do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeleteObject](#) na Referência de AWS CLI Comandos.

describe-object

O código de exemplo a seguir mostra como usar `describe-object`.

AWS CLI

Para ver os cabeçalhos de um objeto

O `describe-object` exemplo a seguir exibe os cabeçalhos de um objeto no caminho especificado.

```
aws mediastore-data describe-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg
```

Saída:

```
{
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
  "ContentType": "image/jpeg",
  "ContentLength": "3860266",
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f555555555555da6d3"
}
```

Para obter mais informações, consulte [Visualizando os detalhes de um objeto](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [DescribeObject](#) na Referência de AWS CLI Comandos.

get-object

O código de exemplo a seguir mostra como usar `get-object`.

AWS CLI

Exemplo 1: Para baixar um objeto inteiro

O `get-object` exemplo a seguir baixa o objeto especificado.

```
aws mediastore-data get-object \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \
  --path events/baseball/setup.jpg setup.jpg
```

Saída:

```
{
  "ContentType": "image/jpeg",
  "StatusCode": 200,
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f555555555555da6d3",
  "ContentLength": "3860266",
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"
}
```

Exemplo 2: Para baixar parte de um objeto

O `get-object` exemplo a seguir baixa a parte especificada de um objeto.

```
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"
```

Saída:

```
{  
  "StatusCode": 206,  
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",  
  "ContentType": "image/jpeg",  
  "ContentRange": "bytes 0-100/3860266",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",  
  "ContentLength": "101"  
}
```

Para obter mais informações, consulte [Baixar um objeto](#) no Guia do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [GetObject](#) na Referência de AWS CLI Comandos.

list-items

O código de exemplo a seguir mostra como usar `list-items`.

AWS CLI

Exemplo 1: Para visualizar uma lista de itens (objetos e pastas) armazenados em um contêiner

O `list-items` exemplo a seguir exibe uma lista de itens (objetos e pastas) armazenados no contêiner especificado.

```
aws mediastore-data list-items \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com
```

Saída:

```
{  
  "Items": [  
    {  
      "Type": "OBJECT",
```



```
"ContentSHA256":  
"f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",  
"StorageClass": "TEMPORAL"  
}
```

Para obter mais informações, consulte [Carregando um objeto no Guia](#) do MediaStore usuário do AWS Elemental.

- Para API obter detalhes, consulte [PutObject](#) na Referência de AWS CLI Comandos.

MediaTailor exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MediaTailor.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-playback-configuration

O código de exemplo a seguir mostra como usar delete-playback-configuration.

AWS CLI

Para excluir uma configuração

O seguinte delete-playback-configuration exclui uma configuração chamada `campaign_short`.

```
aws mediatailor delete-playback-configuration \  
--name campaign_short
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo uma configuração no Guia](#) do MediaTailor usuário do AWS Elemental.

- Para API obter detalhes, consulte [DeletePlaybackConfiguration](#) na Referência de AWS CLI Comandos.

get-playback-configuration

O código de exemplo a seguir mostra como usar `get-playback-configuration`.

AWS CLI

Para descrever uma configuração

A seguir, são `get-playback-configuration` exibidos todos os detalhes da configuração nomeada `west_campaign`.

```
aws mediatailor get-playback-configuration \  
  --name west_campaign
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Visualizando uma configuração](#) no Guia do MediaTailor usuário do AWS Elemental.

- Para API obter detalhes, consulte [GetPlaybackConfiguration](#) na Referência de AWS CLI Comandos.

list-playback-configurations

O código de exemplo a seguir mostra como usar `list-playback-configurations`.

AWS CLI

Para listar todas as configurações

A seguir, são `list-playback-configurations` exibidos todos os detalhes da configuração na AWS conta atual.

```
aws mediatailor list-playback-configurations
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte Visualizando uma configuração < <https://docs.aws.amazon.com/mediatailor/latest/ug/configurations-view.html> > no Guia do usuário do Elemental.AWS MediaTailor

- Para API obter detalhes, consulte [ListPlaybackConfigurations](#) na Referência de AWS CLI Comandos.

put-playback-configuration

O código de exemplo a seguir mostra como usar `put-playback-configuration`.

AWS CLI

Para criar uma configuração

O seguinte `put-playback-configuration` cria uma configuração chamada `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

```

Saída:

```

{
  "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"
}
```

Para obter mais informações, consulte [Criação de uma configuração](#) no Guia do MediaTailor usuário do AWS Elemental.

- Para API obter detalhes, consulte [PutPlaybackConfiguration](#) na Referência de AWS CLI Comandos.

Exemplos de MemoryDB usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with MemoryDB.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

copy-snapshot

O código de exemplo a seguir mostra como usar `copy-snapshot`.

AWS CLI

Para copiar um instantâneo

O `copy-snapshot` exemplo a seguir cria uma cópia de um snapshot.

```
aws memorydb copy-snapshot \  
  --source-snapshot-name my-cluster-snapshot \  
  --target-snapshot-name my-cluster-snapshot-copy
```

Saída

```
{  
  "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  
    }  
  }  
}
```

Para obter mais informações, consulte Como [copiar um snapshot no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [CopySnapshot](#) na Referência de AWS CLI Comandos.

create-acl

O código de exemplo a seguir mostra como usar `create-acl`.

AWS CLI

Para criar um ACL

O `create-acl` exemplo a seguir cria uma nova lista de controle de acesso.

```
aws memorydb create-acl \  
  --acl-name "new-acl-1" \  
  --user-names "my-user"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [CreateAcl](#) na Referência de AWS CLI Comandos.

create-cluster

O código de exemplo a seguir mostra como usar `create-cluster`.

AWS CLI

Para criar um cluster

O `create-cluster` exemplo a seguir cria um novo cluster.

```
aws memorydb create-cluster \  
  --cluster-name my-new-cluster \  
  --node-type db.r6g.large \  
  --acl-name my-acl \  
  --subnet-group my-sg
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Gerenciando clusters](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [CreateCluster](#) na Referência de AWS CLI Comandos.

create-parameter-group

O código de exemplo a seguir mostra como usar `create-parameter-group`.

AWS CLI

Para criar um grupo de parâmetros

O `create-parameter-group` exemplo a seguir cria um grupo de parâmetros.

```
aws memorydb create-parameter-group \  
  --parameter-group-name myRedis6x \  
  --family memorydb_redis6 \  
  --description "my-parameter-group"
```

Saída:

```
{  
  "ParameterGroup": {  
    "Name": "myredis6x",  
    "Family": "memorydb_redis6",  
    "Description": "my-parameter-group",  
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"  
  }  
}
```

Para obter mais informações, consulte [Criação de um grupo de parâmetros](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [CreateParameterGroup](#) na Referência de AWS CLI Comandos.

create-snapshot

O código de exemplo a seguir mostra como usar `create-snapshot`.

AWS CLI

Para criar um instantâneo

O `create-snapshot` exemplo a seguir cria um instantâneo.

```
aws memorydb create-snapshot \  
  --parameter-group-name myRedis6x \  
  --snapshot-name myRedis6x-snapshot \  
  --description "my-snapshot"
```

```
--cluster-name my-cluster \  
--snapshot-name my-cluster-snapshot
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

Para obter mais informações, consulte Como [criar instantâneos manuais](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [CreateSnapshot](#) na Referência de AWS CLI Comandos.

create-subnet-group

O código de exemplo a seguir mostra como usar create-subnet-group.

AWS CLI

Para criar um grupo de sub-redes

O create-subnet-group exemplo a seguir cria um grupo de sub-redes.

```
aws memorydb create-subnet-group \  
  --subnet-group-name mysubnetgroup \  
  --description "my subnet group" \  
  --subnet-ids subnet-5623xxxx
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de um grupo de sub-redes no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [CreateSubnetGroup](#) na Referência de AWS CLI Comandos.

create-user

O código de exemplo a seguir mostra como usar create-user.

AWS CLI

Para criar um usuário

O create-user exemplo a seguir cria um novo usuário.

```
aws memorydb create-user \  
  --user-name user-name-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --
```

```
--authentication-mode \  
    Passwords="enterapasswordhere",Type=password
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

delete-acl

O código de exemplo a seguir mostra como usar delete-acl.

AWS CLI

Para excluir um ACL

O delete-acl exemplo a seguir exclui uma lista de controle de acesso.

```
aws memorydb delete-acl \  
    --acl-name "new-acl-1"
```

Saída:

```
{
```

```

    "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"
    }
  }
}

```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DeleteAcl](#) na Referência de AWS CLI Comandos.

delete-cluster

O código de exemplo a seguir mostra como usar delete-cluster.

AWS CLI

Para excluir um cluster

O delete-cluster exemplo a seguir exclui um cluster.

```

aws memorydb delete-cluster \
  --cluster-name my-new-cluster

```

Saída:

```

{
  "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
  }
}

```

Para obter mais informações, consulte [Excluindo um cluster](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DeleteCluster](#) na Referência de AWS CLI Comandos.

delete-parameter-group

O código de exemplo a seguir mostra como usar delete-parameter-group.

AWS CLI

Para excluir um grupo de parâmetros

O delete-parameter-group exemplo a seguir exclui um grupo de parâmetros.

```

aws memorydb delete-parameter-group \
  --parameter-group-name myRedis6x

```

Saída:

```

{
  "ParameterGroup": {
    "Name": "myredis6x",
    "Family": "memorydb_redis6",
    "Description": "my-parameter-group",
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"
  }
}

```

Para obter mais informações, consulte [Excluindo um grupo de parâmetros no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DeleteParameterGroup](#) na Referência de AWS CLI Comandos.

delete-snapshot

O código de exemplo a seguir mostra como usar delete-snapshot.

AWS CLI

Para excluir um snapshot

O delete-snapshot exemplo a seguir exclui um instantâneo.

```
aws memorydb delete-snapshot \  
  --snapshot-name my-cluster-snapshot
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

```
}  
}
```

Para obter mais informações, consulte [Excluindo um snapshot](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DeleteSnapshot](#) na Referência de AWS CLI Comandos.

delete-subnet-group

O código de exemplo a seguir mostra como usar `delete-subnet-group`.

AWS CLI

Para excluir um grupo de sub-redes

O `delete-subnet-group` exemplo a seguir exclui uma sub-rede.

```
aws memorydb delete-subnet-group \  
  --subnet-group-name mysubnetgroup
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Excluindo um grupo de sub-redes](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DeleteSubnetGroup](#) na Referência de AWS CLI Comandos.

delete-user

O código de exemplo a seguir mostra como usar delete-user.

AWS CLI

Como excluir um usuário

O delete-user exemplo a seguir exclui um usuário.

```
aws memorydb delete-user \  
  --user-name my-user
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

describe-acls

O código de exemplo a seguir mostra como usar describe-acls.

AWS CLI

Para retornar uma lista de ACLs

O `describe-acls` a seguir retorna uma lista de ACLs

```
aws memorydb describe-acls
```

Saída:

```
{
  "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-1:49165xxxxxx:acl/my-acl"
    }
  ]
}
```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeAcls](#) na Referência de AWS CLI Comandos.

`describe-clusters`

O código de exemplo a seguir mostra como usar `describe-clusters`.

AWS CLI

Para retornar uma lista de clusters

O seguinte `describe-clusters`` retorna uma lista de clusters.

```
aws memorydb describe-clusters
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de clusters](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeClusters](#) na Referência de AWS CLI Comandos.

describe-engine-versions

O código de exemplo a seguir mostra como usar `describe-engine-versions`.

AWS CLI

Para retornar uma lista das versões do mecanismo

O `describe-engine-versions` a seguir retorna uma lista das versões do mecanismo.

```
aws memorydb describe-engine-versions
```

Saída:

```
{
  "EngineVersions": [
    {
      "EngineVersion": "6.2",
      "EnginePatchVersion": "6.2.6",
      "ParameterGroupFamily": "memorydb_redis6"
    }
  ]
}
```

Para obter mais informações, consulte [Versões e atualizações do mecanismo no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeEngineVersions](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para retornar uma lista de eventos

O comando `describe-events`` a seguir retorna uma lista de eventos.

```
aws memorydb describe-events
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Monitoramento de eventos](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-parameter-groups

O código de exemplo a seguir mostra como usar `describe-parameter-groups`.

AWS CLI

Para retornar uma lista de grupos de parâmetros

O `describe-parameter-groups`` a seguir retorna uma lista de grupos de parâmetros.

```
aws memorydb describe-parameter-groups
```

Saída:


```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Configurando parâmetros do mecanismo usando grupos de parâmetros no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeParameterGroups](#) na Referência de AWS CLI Comandos.

describe-parameters

O código de exemplo a seguir mostra como usar describe-parameters.

AWS CLI

Para retornar uma lista de parâmetros

O comando describe-parameters` a seguir retorna uma lista de parâmetros.

```
aws memorydb describe-parameters
```

Saída:

```
{
  "Parameters": [
    {
      "Name": "acllog-max-len",
      "Value": "128",
      "Description": "The maximum length of the ACL Log",
      "DataType": "integer",
      "AllowedValues": "1-10000",
      "MinimumEngineVersion": "6.2.4"
    },
  ],
}
```

```
{
  "Name": "activedefrag",
  "Value": "no",
  "Description": "Enabled active memory defragmentation",
  "DataType": "string",
  "AllowedValues": "yes,no",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "active-defrag-cycle-max",
  "Value": "75",
  "Description": "Maximal effort for defrag in CPU percentage",
  "DataType": "integer",
  "AllowedValues": "1-75",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "active-defrag-cycle-min",
  "Value": "5",
  "Description": "Minimal effort for defrag in CPU percentage",
  "DataType": "integer",
  "AllowedValues": "1-75",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "active-defrag-ignore-bytes",
  "Value": "104857600",
  "Description": "Minimum amount of fragmentation waste to start active
defrag",
  "DataType": "integer",
  "AllowedValues": "1048576-",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "active-defrag-max-scan-fields",
  "Value": "1000",
  "Description": "Maximum number of set/hash/zset/list fields that will be
processed from the main dictionary scan",
  "DataType": "integer",
  "AllowedValues": "1-1000000",
  "MinimumEngineVersion": "6.2.4"
},
{
  "Name": "active-defrag-threshold-lower",
```

```

    "Value": "10",
    "Description": "Minimum percentage of fragmentation to start active
defrag",
    "DataType": "integer",
    "AllowedValues": "1-100",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-defrag-threshold-upper",
    "Value": "100",
    "Description": "Maximum percentage of fragmentation at which we use
maximum effort",
    "DataType": "integer",
    "AllowedValues": "1-100",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "active-expire-effort",
    "Value": "1",
    "Description": "The amount of effort that redis uses to expire items in
the active expiration job",
    "DataType": "integer",
    "AllowedValues": "1-10",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "activeresharding",
    "Value": "yes",
    "Description": "Apply rehashing or not",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-normal-hard-limit",
    "Value": "0",
    "Description": "Normal client output buffer hard limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-normal-soft-limit",
    "Value": "0",

```

```
    "Description": "Normal client output buffer soft limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-normal-soft-seconds",
    "Value": "0",
    "Description": "Normal client output buffer soft limit in seconds",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-pubsub-hard-limit",
    "Value": "33554432",
    "Description": "Pubsub client output buffer hard limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-pubsub-soft-limit",
    "Value": "8388608",
    "Description": "Pubsub client output buffer soft limit in bytes",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "client-output-buffer-limit-pubsub-soft-seconds",
    "Value": "60",
    "Description": "Pubsub client output buffer soft limit in seconds",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hash-max-ziplist-entries",
    "Value": "512",
    "Description": "The maximum number of hash entries in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
```

```
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hash-max-ziplist-value",
    "Value": "64",
    "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "hll-sparse-max-bytes",
    "Value": "3000",
    "Description": "HyperLogLog sparse representation bytes limit",
    "DataType": "integer",
    "AllowedValues": "1-16000",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-eviction",
    "Value": "no",
    "Description": "Perform an asynchronous delete on evictions",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-expire",
    "Value": "no",
    "Description": "Perform an asynchronous delete on expired keys",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "lazyfree-lazy-server-del",
    "Value": "no",
    "Description": "Perform an asynchronous delete on key updates",
    "DataType": "string",
    "AllowedValues": "yes,no",
    "MinimumEngineVersion": "6.2.4"
  },
  {
```

```

        "Name": "lazyfree-lazy-user-del",
        "Value": "no",
        "Description": "Specifies whether the default behavior of DEL command
acts the same as UNLINK",
        "DataType": "string",
        "AllowedValues": "yes,no",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "lfu-decay-time",
        "Value": "1",
        "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "lfu-log-factor",
        "Value": "10",
        "Description": "The log factor for incrementing key counter for LFU
eviction policy",
        "DataType": "integer",
        "AllowedValues": "1-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "list-compress-depth",
        "Value": "0",
        "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
        "DataType": "integer",
        "AllowedValues": "0-",
        "MinimumEngineVersion": "6.2.4"
    },
    {
        "Name": "maxmemory-policy",
        "Value": "noeviction",
        "Description": "Max memory policy",
        "DataType": "string",
        "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
        "MinimumEngineVersion": "6.2.4"
    }

```

```
  },
  {
    "Name": "maxmemory-samples",
    "Value": "3",
    "Description": "Max memory samples",
    "DataType": "integer",
    "AllowedValues": "1-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "notify-keyspace-events",
    "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
    "DataType": "string",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "set-max-intset-entries",
    "Value": "512",
    "Description": "The limit in the size of the set in order for the
dataset to be compressed",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "slowlog-log-slower-than",
    "Value": "10000",
    "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command",
    "DataType": "integer",
    "AllowedValues": "-",
    "MinimumEngineVersion": "6.2.4"
  },
  {
    "Name": "slowlog-max-len",
    "Value": "128",
    "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
    "DataType": "integer",
    "AllowedValues": "0-",
    "MinimumEngineVersion": "6.2.4"
  }
}
```

```
    },
    {
      "Name": "stream-node-max-bytes",
      "Value": "4096",
      "Description": "The maximum size of a single node in a stream in bytes",
      "DataType": "integer",
      "AllowedValues": "0-",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "stream-node-max-entries",
      "Value": "100",
      "Description": "The maximum number of items a single node in a stream
can contain",
      "DataType": "integer",
      "AllowedValues": "0-",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "tcp-keepalive",
      "Value": "300",
      "Description": "If non-zero, send ACKs every given number of seconds",
      "DataType": "integer",
      "AllowedValues": "0-",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "timeout",
      "Value": "0",
      "Description": "Close connection if client is idle for a given number of
seconds, or never if 0",
      "DataType": "integer",
      "AllowedValues": "0,20-",
      "MinimumEngineVersion": "6.2.4"
    },
    {
      "Name": "tracking-table-max-keys",
      "Value": "1000000",
      "Description": "The maximum number of keys allowed for the tracking
table for client side caching",
      "DataType": "integer",
      "AllowedValues": "1-1000000000",
      "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"
    }
  ]
}

```

Para obter mais informações, consulte [Configurando parâmetros do mecanismo usando grupos de parâmetros no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeParameters](#) na Referência de AWS CLI Comandos.

describe-snapshots

O código de exemplo a seguir mostra como usar describe-snapshots.

AWS CLI

Para retornar uma lista de instantâneos

O seguinte describe-snapshots` retorna uma lista de instantâneos.

```
aws memorydb describe-snapshots
```

Saída:

```

{
  "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
    }
  }
}
```

Para obter mais informações, consulte [Snapshot and restore](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeSnapshots](#) na Referência de AWS CLI Comandos.

describe-subnet-groups

O código de exemplo a seguir mostra como usar `describe-subnet-groups`.

AWS CLI

Para retornar uma lista de grupos de sub-redes

O `describe-subnet-groups` a seguir retorna uma lista de grupos de sub-redes.

```
aws memorydb describe-subnet-groups
```

Saída

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeSubnetGroups](#) na Referência de AWS CLI Comandos.

describe-users

O código de exemplo a seguir mostra como usar `describe-users`.

AWS CLI

Para retornar uma lista de usuários

O comando `describe-users` a seguir retorna uma lista de usuários.

```
aws memorydb describe-users
```

Saída

```
{
```

```

"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"
  }
]
}

```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [DescribeUsers](#) na Referência de AWS CLI Comandos.

failover-shard

O código de exemplo a seguir mostra como usar failover-shard.

AWS CLI

Falhar em um fragmento

O seguinte failover-shard` falha em um fragmento.

```
aws memorydb failover-shard \  
  --cluster-name my-cluster --shard-name 0001
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Minimizando o tempo de inatividade com o MultiAZ](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [FailoverShard](#) na Referência de AWS CLI Comandos.

list-allowed-node-type-updates

O código de exemplo a seguir mostra como usar `list-allowed-node-type-updates`.

AWS CLI

Para retornar uma lista de atualizações de tipo de nó permitidas

Os seguintes `list-allowed-node-type-updates` retornam uma lista de atualizações de tipos de nós disponíveis.

```
aws memorydb list-allowed-node-type-updates
```

Saída:

```
{
  "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
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Scaling no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [ListAllowedNodeTypeUpdates](#) na Referência de AWS CLI Comandos.

list-tags

O código de exemplo a seguir mostra como usar `list-tags`.

AWS CLI

Para retornar uma lista de tags

As tags de lista a seguir retornam uma lista de tags.

```
aws memorydb list-tags \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster
```

Saída:

```
{  
  "TagList": [  
    {  
      "Key": "mytag",  
      "Value": "myvalue"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Recursos de marcação no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [ListTags](#) na Referência de AWS CLI Comandos.

reset-parameter-group

O código de exemplo a seguir mostra como usar `reset-parameter-group`.

AWS CLI

Para redefinir um grupo de parâmetros

O seguinte `reset-parameter-group` redefina um grupo de parâmetros.

```
aws memorydb reset-parameter-group \  
  --parameter-group-name my-parameter-group \  
  --all-parameters
```

Saída:

```
{  
  "ParameterGroup": {  
    "Name": "my-parameter-group",  
    "Family": "memorydb_redis6",  
    "Description": "my parameter group",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-  
group"  
  }  
}
```

Para obter mais informações, consulte [Configurando parâmetros do mecanismo usando grupos de parâmetros no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [ResetParameterGroup](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um recurso

A seguinte `tag-resource` adiciona uma tag a um recurso.

```
aws memorydb tag-resource \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster \  
  --tags Key="mykey",Value="myvalue"
```

Saída:

```
{  
  "TagList": [  
    {  
      "Key": "mykey",  
      "Value": "myvalue"  
    }  
  ]  
}
```



```

    {
      "Key": "mytag",
      "Value": "myvalue"
    },
    {
      "Key": "mykey",
      "Value": "myvalue"
    }
  ]
}

```

Para obter mais informações, consulte [Recursos de marcação no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para atualizar um ACL

As seguintes atualizações de `update-acl` podem ser adicionadas a um usuárioACL.

```

aws memorydb untag-resource \
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \
  --tag-keys mykey

```

Saída:

```

{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    }
  ]
}

```

Para obter mais informações, consulte [Recursos de marcação no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-cluster

O código de exemplo a seguir mostra como usar `update-cluster`.

AWS CLI

Para atualizar um cluster

O seguinte `update-cluster` atualiza o grupo de parâmetros de um cluster para `my-parameter-group`

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --parameter-group-name my-parameter-group
```

Saída:

```
{  
  "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
  }
}

```

Para obter mais informações, consulte [Como modificar um cluster](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [UpdateCluster](#) na Referência de AWS CLI Comandos.

update-parameter-group

O código de exemplo a seguir mostra como usar `update-parameter-group`.

AWS CLI

Para atualizar um grupo de parâmetros

O seguinte `update-parameter-group` atualiza um grupo de parâmetros.

```

aws memorydb update-parameter-group \
  --parameter-group-name my-parameter-group \
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"

```

Saída:

```

{
  "ParameterGroup": {
    "Name": "my-parameter-group",
    "Family": "memorydb_redis6",
    "Description": "my parameter group",
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
  }
}

```

Para obter mais informações, consulte [Modificação de um grupo de parâmetros no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [UpdateParameterGroup](#) na Referência de AWS CLI Comandos.

update-subnet-group

O código de exemplo a seguir mostra como usar `update-subnet-group`.

AWS CLI

Para atualizar um grupo de sub-redes

O seguinte `update-subnet-group` atualiza o ID de sub-rede de um grupo de sub-redes.

```
aws memorydb update-subnet-group \  
  --subnet-group-name my-sg \  
  --subnet-ids subnet-01f29d458f3xxxxxx
```

Saída:

```
{  
  "SubnetGroup": {  
    "Name": "my-sg-1",  
    "Description": "my-sg",  
    "VpcId": "vpc-09d2cfc01xxxxxxxx",  
    "Subnets": [  
      {  
        "Identifier": "subnet-01f29d458f3xxxxxx",  
        "AvailabilityZone": {  
          "Name": "us-east-1a"  
        }  
      }  
    ],  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"  
  }  
}
```

Para obter mais informações, consulte [Sub-redes e grupos de sub-redes](#) no Guia do usuário do MemoryDB.

- Para API obter detalhes, consulte [UpdateSubnetGroup](#) na Referência de AWS CLI Comandos.

update-user

O código de exemplo a seguir mostra como usar `update-user`.

AWS CLI

Para atualizar um usuário

O seguinte `update-user` modifica a string de acesso de um usuário.

```
aws memorydb update-user \  
  --user-name my-user \  
  --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Autenticação de usuários com listas de controle de acesso no Guia](#) do usuário do MemoryDB.

- Para API obter detalhes, consulte [UpdateUser](#) na Referência de AWS CLI Comandos.

MSK Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonMSK.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-cluster

O código de exemplo a seguir mostra como usar `create-cluster`.

AWS CLI

Para criar um MSK cluster da Amazon

O `create-cluster` exemplo a seguir cria um MSK cluster chamado `MessagingCluster` com três nós de agente. Um JSON arquivo chamado `brokernodegroupinfo.json` especifica as três sub-redes pelas quais você deseja que a Amazon MSK distribua os nós do broker. Este exemplo não especifica o nível de monitoramento, então o cluster obtém o `DEFAULT` nível.

```
aws kafka create-cluster \  
  --cluster-name "MessagingCluster" \  
  --broker-node-group-info file://brokernodegroupinfo.json \  
  --kafka-version "2.2.1" \  
  --number-of-broker-nodes 3
```

Conteúdo de `brokernodegroupinfo.json`:

```
{  
  "InstanceType": "kafka.m5.xlarge",  
  "BrokerAZDistribution": "DEFAULT",  
  "ClientSubnets": [  
    "subnet-0123456789111abcd",  
    "subnet-0123456789222abcd",  
    "subnet-0123456789333abcd"  
  ]  
}
```

Saída:

```
{
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "ClusterName": "MessagingCluster",
  "State": "CREATING"
}
```

Para obter mais informações, consulte [Criar um MSK cluster da Amazon](#) no Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [CreateCluster](#) na Referência de AWS CLI Comandos.

create-configuration

O código de exemplo a seguir mostra como usar create-configuration.

AWS CLI

Para criar uma MSK configuração personalizada da Amazon

O create-configuration exemplo a seguir cria uma MSK configuração personalizada com as propriedades do servidor especificadas no arquivo de entrada.

```
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
```

Conteúdo de configuration.txt:

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

Este comando não produz saída. Saída:

```
{
  "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"
  }

```

Para obter mais informações, consulte [Operações de MSK configuração da Amazon](#) no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [CreateConfiguration](#) na Referência de AWS CLI Comandos.

describe-cluster

O código de exemplo a seguir mostra como usar `describe-cluster`.

AWS CLI

Para descrever um cluster

O `describe-cluster` exemplo a seguir descreve um MSK cluster da Amazon.

```

aws kafka describe-cluster \
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5

```

Saída:

```

{
  "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"
  }
}
```

```
}
```

Para obter mais informações, consulte [Listar MSK clusters da Amazon](#) no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [DescribeCluster](#) na Referência de AWS CLI Comandos.

get-bootstrap-brokers

O código de exemplo a seguir mostra como usar `get-bootstrap-brokers`.

AWS CLI

Para obter corretores de bootstrap

O `get-bootstrap-brokers` exemplo a seguir recupera as informações do agente de bootstrap para um cluster da 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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Como obter os Bootstrap Brokers](#) no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [GetBootstrapBrokers](#) na Referência de AWS CLI Comandos.

list-clusters

O código de exemplo a seguir mostra como usar `list-clusters`.

AWS CLI

Para listar os clusters disponíveis

O `list-clusters` exemplo a seguir lista os MSK clusters da Amazon em sua AWS conta.

```
aws kafka list-clusters
```

Saída:

```
{
  "ClusterInfoList": [
    {
      "BrokerNodeGroupInfo": {
        "BrokerAZDistribution": "DEFAULT",
        "ClientSubnets": [
          "subnet-cbfff283",
          "subnet-6746046b"
        ],
        "InstanceType": "kafka.m5.large",
        "SecurityGroups": [
          "sg-f839b688"
        ],
        "StorageInfo": {
          "EbsStorageInfo": {
            "VolumeSize": 100
          }
        }
      },
      "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
      "ClusterName": "demo-cluster-1",
      "CreationTime": "2020-07-09T02:31:36.223000+00:00",
      "CurrentBrokerSoftwareInfo": {
        "KafkaVersion": "2.2.1"
      },
      "CurrentVersion": "K3AEGXETSR30VB",
      "EncryptionInfo": {
        "EncryptionAtRest": {
          "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a7ca56d5-0768-4b64-a670-339a9fbef81c"
        },
        "EncryptionInTransit": {
```

```

        "ClientBroker": "TLS_PLAINTEXT",
        "InCluster": true
    }
},
"EnhancedMonitoring": "DEFAULT",
"OpenMonitoring": {
    "Prometheus": {
        "JmxExporter": {
            "EnabledInBroker": false
        },
        "NodeExporter": {
            "EnabledInBroker": false
        }
    }
},
"NumberOfBrokerNodes": 2,
"State": "ACTIVE",
"Tags": {},
"ZookeeperConnectionString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
    }
]
}

```

Para obter mais informações, consulte [Listar MSK clusters da Amazon](#) no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [ListClusters](#) na Referência de AWS CLI Comandos.

update-broker-storage

O código de exemplo a seguir mostra como usar `update-broker-storage`.

AWS CLI

Para atualizar o EBS armazenamento para corretores

O `update-broker-storage` exemplo a seguir atualiza a quantidade de EBS armazenamento de todos os agentes no cluster. A Amazon MSK define o valor de armazenamento alvo para cada corretor de acordo com o valor especificado no exemplo. Você pode obter a versão atual do cluster descrevendo o cluster ou listando todos os clusters.

```
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"
```

A saída retorna an ARN para essa update-broker-storage operação. Para determinar se essa operação foi concluída, use o describe-cluster-operation comando com isso ARN como entrada.

```
{
  "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"
}
```

Para obter mais informações, consulte [Atualizar o EBS armazenamento para corretores](#) no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [UpdateBrokerStorage](#) na Referência de AWS CLI Comandos.

update-cluster-configuration

O código de exemplo a seguir mostra como usar update-cluster-configuration.

AWS CLI

Para atualizar a configuração de um MSK cluster da Amazon

O update-cluster-configuration exemplo a seguir atualiza a configuração do MSK cluster existente especificado. Ele usa uma MSK configuração personalizada.

```
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"
```

Conteúdo de configuration-info.json:

```
{
  "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "Revision": 1
}
```

A saída retorna an ARN para essa `update-cluster-configuration` operação. Para determinar se essa operação foi concluída, use o `describe-cluster-operation` comando com isso ARN como entrada.

```
{
  "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"
}
```

Para obter mais informações, consulte [Atualizar a configuração de um MSK cluster da Amazon no Guia](#) do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

- Para API obter detalhes, consulte [UpdateClusterConfiguration](#) na Referência de AWS CLI Comandos.

Exemplos do Network Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Network Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-customer-gateway

O código de exemplo a seguir mostra como usar `associate-customer-gateway`.

AWS CLI

Para associar um gateway do cliente

O `associate-customer-gateway` exemplo a seguir associa o gateway do cliente `cgw-11223344556677889` na rede global especificada ao 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-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --region us-west-2
```

Saída:

```
{  
  "CustomerGatewayAssociation": {  
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/  
cgw-11223344556677889",  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "State": "PENDING"  
  }  
}
```

Para obter mais informações, consulte [Customer Gateway Associations](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [AssociateCustomerGateway](#) na Referência de AWS CLI Comandos.

associate-link

O código de exemplo a seguir mostra como usar `associate-link`.

AWS CLI

Para associar um link

O `associate-link` exemplo a seguir associa o link `link-11112222aaaabbbb1` ao dispositivo `device-07f6fd08867abc123`. O link e o dispositivo estão na rede global especificada.

```
aws networkmanager associate-link \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

Saída:

```
{  
  "LinkAssociation": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkAssociationState": "PENDING"  
  }  
}
```

Para obter mais informações, consulte [Associações de dispositivos e links](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [AssociateLink](#) na Referência de AWS CLI Comandos.

create-core-network

O código de exemplo a seguir mostra como usar `create-core-network`.

AWS CLI

Para criar uma rede principal

O `create-core-network` exemplo a seguir cria uma rede principal usando uma descrição e tags opcionais em uma rede WAN global do AWS Cloud.


```
aws networkmanager create-core-network \  
  --global-network-id global-network-0d59060f16a73bc41 \  
  --description "Main headquarters location" \  
  --tags Key=Name,Value="New York City office"
```

Saída:

```
{  
  "CoreNetwork": {  
    "GlobalNetworkId": "global-network-0d59060f16a73bc41",  
    "CoreNetworkId": "core-network-0fab62fe438d94db6",  
    "CoreNetworkArn": "arn:aws:networkmanager::987654321012:core-network/core-network-0fab62fe438d94db6",  
    "Description": "Main headquarters location",  
    "CreatedAt": "2022-01-10T19:53:59+00:00",  
    "State": "AVAILABLE",  
    "Tags": [  
      {  
        "Key": "Name",  
        "Value": "New York City office"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte o Guia do WAN usuário de [redes principais](#) no AWS Cloud.

- Para API obter detalhes, consulte [CreateCoreNetwork](#) na Referência de AWS CLI Comandos.

create-device

O código de exemplo a seguir mostra como usar create-device.

AWS CLI

Para criar um dispositivo

O create-device exemplo a seguir cria um dispositivo na rede global especificada. Os detalhes do dispositivo incluem uma descrição, o tipo, o fornecedor, o modelo e o número de série.

```
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
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Trabalhando com dispositivos](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [CreateDevice](#) na Referência de AWS CLI Comandos.

create-global-network

O código de exemplo a seguir mostra como usar `create-global-network`.

AWS CLI

Para criar uma rede global

Os `create-global-network` exemplos a seguir criam uma nova rede global. O estado inicial após a criação é `PENDING`.

```
aws networkmanager create-global-network
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateGlobalNetwork](#) na Referência de AWS CLI Comandos.

create-link

O código de exemplo a seguir mostra como usar `create-link`.

AWS CLI

Para criar um link

O `create-link` exemplo a seguir cria um link na rede global especificada. O link inclui uma descrição e detalhes sobre o tipo de link, a largura de banda e o provedor. O ID do site indica o site ao qual o link está associado.

```
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
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Trabalhando com links](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [CreateLink](#) na Referência de AWS CLI Comandos.

create-site

O código de exemplo a seguir mostra como usar `create-site`.

AWS CLI

Para criar um site

O `create-site` exemplo a seguir cria um site na rede global especificada. Os detalhes do site incluem uma descrição e as informações de localização.

```
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
```

Saída:

```
{
```

```

"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"
}
}

```

Para obter mais informações, consulte [Trabalhando com sites](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [CreateSite](#) na Referência de AWS CLI Comandos.

create-vpc-attachment

O código de exemplo a seguir mostra como usar create-vpc-attachment.

AWS CLI

Para criar um VPC anexo

O create-vpc-attachment exemplo a seguir cria um VPC anexo com IPv6 suporte em uma rede principal.

```

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

```

Saída:

```

{
  "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
    }
  }
}

```

Para obter mais informações, consulte [Criar um anexo](#) no Cloud WAN User Guide.

- Para API obter detalhes, consulte [CreateVpcAttachment](#) na Referência de AWS CLI Comandos.

delete-attachment

O código de exemplo a seguir mostra como usar delete-attachment.

AWS CLI

Para excluir um anexo

O delete-attachment exemplo a seguir exclui um anexo do Connect.

```

aws networkmanager delete-attachment \
  --attachment-id attachment-01feddaeae26ab68c

```

Saída:

```
{
```

```
"Attachment": {
  "CoreNetworkId": "core-network-0f4b0a9d5ee7761d1",
  "AttachmentId": "attachment-01feddaeae26ab68c",
  "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"
}
```

Para obter mais informações, consulte [Excluir anexos](#) no Cloud WAN User Guide.

- Para API obter detalhes, consulte [DeleteAttachment](#) na Referência de AWS CLI Comandos.

delete-bucket-analytics-configuration

O código de exemplo a seguir mostra como usar delete-bucket-analytics-configuration.

AWS CLI

Para excluir uma configuração de análise de um bucket

O exemplo delete-bucket-analytics-configuration a seguir recupera a configuração referente ao bucket e ID especificados.

```
aws s3api delete-bucket-analytics-configuration \
  --bucket my-bucket \
  --id 1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBucketAnalyticsConfiguration](#) na Referência de AWS CLI Comandos.

delete-bucket-metrics-configuration

O código de exemplo a seguir mostra como usar delete-bucket-metrics-configuration.

AWS CLI

Para excluir uma configuração de métricas referente a um bucket

O exemplo `delete-bucket-metrics-configuration` a seguir remove a configuração referente ao bucket e ID especificados.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBucketMetricsConfiguration](#) na Referência de AWS CLI Comandos.

`delete-core-network`

O código de exemplo a seguir mostra como usar `delete-core-network`.

AWS CLI

Para excluir uma rede principal

O `delete-core-network` exemplo a seguir exclui uma rede principal de uma rede WAN global em nuvem.

```
aws networkmanager delete-core-network \  
  --core-network-id core-network-0fab62fe438d94db6
```

Saída:

```
{  
  "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": []
    }
]
}
}

```

Para obter mais informações, consulte o Guia do WAN usuário de [redes principais](#) no Cloud.

- Para API obter detalhes, consulte [DeleteCoreNetwork](#) na Referência de AWS CLI Comandos.

delete-device

O código de exemplo a seguir mostra como usar delete-device.

AWS CLI

Para excluir um dispositivo

O delete-device exemplo a seguir exclui o dispositivo especificado da rede global especificada.

```

aws networkmanager delete-device \
  --global-network-id global-network-01231231231231231 \
  --device-id device-07f6fd08867abc123 \
  --region us-west-2

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Trabalhando com dispositivos](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [DeleteDevice](#) na Referência de AWS CLI Comandos.

delete-global-network

O código de exemplo a seguir mostra como usar `delete-global-network`.

AWS CLI

Para excluir uma rede global

O `delete-global-network` exemplo a seguir exclui uma rede global.

```

aws networkmanager delete-global-network \
  --global-network-id global-network-052bedddccb193b6b

```

Saída:

```

{
  "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"
  }
}

```

```
}
```

- Para API obter detalhes, consulte [DeleteGlobalNetwork](#) na Referência de AWS CLI Comandos.

delete-link

O código de exemplo a seguir mostra como usar delete-link.

AWS CLI

Para excluir um link

O delete-link exemplo a seguir exclui o link especificado da rede global especificada.

```
aws networkmanager delete-link \  
  --global-network-id global-network-01231231231231231 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com links](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [DeleteLink](#) na Referência de AWS CLI Comandos.

delete-public-access-block

O código de exemplo a seguir mostra como usar `delete-public-access-block`.

AWS CLI

Para excluir a configuração do Bloqueio de Acesso Público para um bucket

O exemplo `delete-public-access-block` a seguir remove a configuração do Bloqueio de Acesso Público no bucket especificado.

```
aws s3api delete-public-access-block \  
  --bucket my-bucket
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePublicAccessBlock](#) na Referência de AWS CLI Comandos.

delete-site

O código de exemplo a seguir mostra como usar `delete-site`.

AWS CLI

Para excluir um site

O `delete-site` exemplo a seguir exclui o site especificado (`site-444555aaabbb11223`) na rede global especificada.

```
aws networkmanager delete-site \  
  --global-network-id global-network-01231231231231231 \  
  --site-id site-444555aaabbb11223 \  
  --region us-west-2
```

Saída:

```
{
```

```

"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"
}
}

```

Para obter mais informações, consulte [Trabalhando com sites](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [DeleteSite](#) na Referência de AWS CLI Comandos.

deregister-transit-gateway

O código de exemplo a seguir mostra como usar `deregister-transit-gateway`.

AWS CLI

Para cancelar o registro de um gateway de trânsito de uma rede global

O `deregister-transit-gateway` exemplo a seguir cancela o registro do gateway de trânsito especificado da rede global especificada.

```

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

```

Saída:

```

{
  "TransitGatewayRegistration": {

```

```
    "GlobalNetworkId": "global-network-01231231231231231",
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
    "State": {
      "Code": "DELETING"
    }
  }
}
```

Para obter mais informações, consulte [Registros do Transit Gateway](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [DeregisterTransitGateway](#) na Referência de AWS CLI Comandos.

describe-global-networks

O código de exemplo a seguir mostra como usar `describe-global-networks`.

AWS CLI

Para descrever suas redes globais

O `describe-global-networks` exemplo a seguir descreve todas as suas redes globais em sua conta.

```
aws networkmanager describe-global-networks \
  --region us-west-2
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [DescribeGlobalNetworks](#) na Referência de AWS CLI Comandos.

disassociate-customer-gateway

O código de exemplo a seguir mostra como usar `disassociate-customer-gateway`.

AWS CLI

Para desassociar um gateway do cliente

O `disassociate-customer-gateway` exemplo a seguir desassocia o gateway do cliente especificado (`cgw-11223344556677889`) da rede global especificada.

```
aws networkmanager disassociate-customer-gateway \
  --global-network-id global-network-01231231231231 \
  --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
  --region us-west-2
```

Saída:

```
{
  "CustomerGatewayAssociation": {
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "State": "DELETING"
  }
}
```

Para obter mais informações, consulte [Customer Gateway Associations](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [DisassociateCustomerGateway](#) na Referência de AWS CLI Comandos.

disassociate-link

O código de exemplo a seguir mostra como usar `disassociate-link`.

AWS CLI

Para desassociar um link

O `disassociate-link` exemplo a seguir desassocia o link especificado do dispositivo `device-07f6fd08867abc123` na rede global especificada.

```
aws networkmanager disassociate-link \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --link-id link-11112222aaaabbbb1 \  
  --region us-west-2
```

Saída:

```
{  
  "LinkAssociation": {  
    "GlobalNetworkId": "global-network-01231231231231231",  
    "DeviceId": "device-07f6fd08867abc123",  
    "LinkId": "link-11112222aaaabbbb1",  
    "LinkAssociationState": "DELETING"  
  }  
}
```

Para obter mais informações, consulte [Associações de dispositivos e links](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [DisassociateLink](#) na Referência de AWS CLI Comandos.

get-bucket-analytics-configuration

O código de exemplo a seguir mostra como usar `get-bucket-analytics-configuration`.

AWS CLI

Para recuperar a configuração de análise de um bucket com um ID específico

O exemplo `get-bucket-analytics-configuration` a seguir recupera a configuração de análise referente ao bucket e ID especificados.


```
aws s3api get-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Saída:

```
{  
  "AnalyticsConfiguration": {  
    "StorageClassAnalysis": {},  
    "Id": "1"  
  }  
}
```

- Para API obter detalhes, consulte [GetBucketAnalyticsConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-metrics-configuration

O código de exemplo a seguir mostra como usar `get-bucket-metrics-configuration`.

AWS CLI

Para recuperar a configuração de métricas de um bucket com um ID específico

O exemplo `get-bucket-metrics-configuration` a seguir exibe a configuração de métricas referente ao bucket e ID especificados.

```
aws s3api get-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Saída:

```
{  
  "MetricsConfiguration": {  
    "Filter": {  
      "Prefix": "logs"  
    },  
    "Id": "123"  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [GetBucketMetricsConfiguration](#) na Referência de AWS CLI Comandos.

get-customer-gateway-associations

O código de exemplo a seguir mostra como usar `get-customer-gateway-associations`.

AWS CLI

Para obter suas associações de gateway de clientes

O `get-customer-gateway-associations` exemplo a seguir obtém as associações de gateway do cliente para a rede global especificada.

```
aws networkmanager get-customer-gateway-associations \
  --global-network-id global-network-01231231231231 \
  --region us-west-2
```

Saída:

```
{
  "CustomerGatewayAssociations": [
    {
      "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-
gateway/cgw-11223344556677889",
      "GlobalNetworkId": "global-network-01231231231231",
      "DeviceId": "device-07f6fd08867abc123",
      "State": "AVAILABLE"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetCustomerGatewayAssociations](#) na Referência de AWS CLI Comandos.

get-devices

O código de exemplo a seguir mostra como usar `get-devices`.

AWS CLI

Para obter seus dispositivos

O `get-devices` exemplo a seguir obtém os dispositivos na rede global especificada.

```
aws networkmanager get-devices \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [GetDevices](#) na Referência de AWS CLI Comandos.

get-link-associations

O código de exemplo a seguir mostra como usar `get-link-associations`.

AWS CLI

Para obter suas associações de links

O `get-link-associations` exemplo a seguir obtém as associações de links na rede global especificada.

```
aws networkmanager get-link-associations \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Saída:

```
{  
  "LinkAssociations": [  
    {  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "DeviceId": "device-07f6fd08867abc123",  
      "LinkId": "link-11112222aaaabbbb1",  
      "LinkAssociationState": "AVAILABLE"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [GetLinkAssociations](#) na Referência de AWS CLI Comandos.

get-links

O código de exemplo a seguir mostra como usar `get-links`.

AWS CLI

Para obter seus links

O `get-links` exemplo a seguir obtém os links na rede global especificada.

```
aws networkmanager get-links \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Saída:

```
{  
  "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"
  }
]
}
```

- Para API obter detalhes, consulte [GetLinks](#) na Referência de AWS CLI Comandos.

get-object-retention

O código de exemplo a seguir mostra como usar `get-object-retention`.

AWS CLI

Como recuperar a configuração de retenção de objetos de um objeto

O exemplo de `get-object-retention` a seguir recupera a configuração de retenção de objetos do objeto especificado.

```
aws s3api get-object-retention \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf
```

Saída:

```
{
  "Retention": {
    "Mode": "GOVERNANCE",
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"
  }
}
```

- Para API obter detalhes, consulte [GetObjectRetention](#) na Referência de AWS CLI Comandos.

get-public-access-block

O código de exemplo a seguir mostra como usar `get-public-access-block`.

AWS CLI

Para definir ou modificar a configuração do Bloqueio de Acesso Público de um bucket

O exemplo `get-public-access-block` a seguir exibe a configuração do Bloqueio de Acesso Público do bucket especificado.

```
aws s3api get-public-access-block --bucket my-bucket
```

Saída:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
    "RestrictPublicBuckets": true
  }
}
```

- Para API obter detalhes, consulte [GetPublicAccessBlock](#) na Referência de AWS CLI Comandos.

get-sites

O código de exemplo a seguir mostra como usar `get-sites`.

AWS CLI

Para obter seus sites

O `get-sites` exemplo a seguir obtém os sites na rede global especificada.

```
aws networkmanager get-sites \
  --global-network-id global-network-01231231231231 \
  --region us-west-2
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetSites](#) na Referência de AWS CLI Comandos.

get-transit-gateway-registrations

O código de exemplo a seguir mostra como usar `get-transit-gateway-registrations`.

AWS CLI

Para obter seus registros de gateway de trânsito

O `get-transit-gateway-registrations` exemplo a seguir obtém os gateways de trânsito registrados na rede global especificada.

```
aws networkmanager get-transit-gateway-registrations \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2
```

Saída:

```
{
  "TransitGatewayRegistrations": [
```

```

    {
      "GlobalNetworkId": "global-network-01231231231231231",
      "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-
gateway/tgw-123abc05e04123abc",
      "State": {
        "Code": "AVAILABLE"
      }
    }
  ]
}

```

- Para API obter detalhes, consulte [GetTransitGatewayRegistrations](#) na Referência de AWS CLI Comandos.

get-vpc-attachment

O código de exemplo a seguir mostra como usar `get-vpc-attachment`.

AWS CLI

Para obter um VPC anexo

O `get-vpc-attachment` exemplo a seguir retorna informações sobre um VPC anexo.

```

aws networkmanager get-vpc-attachment \
  --attachment-id attachment-03b7ea450134787da

```

Saída:

```

{
  "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-cda9dffc"
  ],
  "Options": {
    "Ipv6Support": false
  }
}
}

```

Para obter mais informações, consulte [Anexos no Guia](#) do WANusuário do Cloud.

- Para API obter detalhes, consulte [GetVpcAttachment](#) na Referência de AWS CLI Comandos.

list-bucket-analytics-configurations

O código de exemplo a seguir mostra como usar `list-bucket-analytics-configurations`.

AWS CLI

Como recuperar uma lista de configurações de análise de um bucket

O exemplo `list-bucket-analytics-configurations` a seguir recupera uma lista de configurações de análise do bucket especificado.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Saída:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},

```

```
        "Id": "1"
      }
    ],
    "IsTruncated": false
  }
```

- Para API obter detalhes, consulte [ListBucketAnalyticsConfigurations](#) na Referência de AWS CLI Comandos.

list-bucket-metrics-configurations

O código de exemplo a seguir mostra como usar `list-bucket-metrics-configurations`.

AWS CLI

Para recuperar uma lista de configurações de métricas para um bucket

O `list-bucket-metrics-configurations` exemplo a seguir recupera uma lista de configurações de métricas para o bucket especificado.

```
aws s3api list-bucket-metrics-configurations \
  --bucket my-bucket
```

Saída:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [ListBucketMetricsConfigurations](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags do recurso de dispositivo especificado (`device-07f6fd08867abc123`).

```
aws networkmanager list-tags-for-resource \
  --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231231/device-07f6fd08867abc123 \
  --region us-west-2
```

Saída:

```
{
  "TagList": [
    {
      "Key": "Network",
      "Value": "Northeast"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-bucket-metrics-configuration

O código de exemplo a seguir mostra como usar `put-bucket-metrics-configuration`.

AWS CLI

Para definir uma configuração de métricas para um bucket

O `put-bucket-metrics-configuration` exemplo a seguir define uma configuração métrica com ID 123 para o bucket especificado.

```
aws s3api put-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123 \  
  --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketMetricsConfiguration](#) na Referência de AWS CLI Comandos.

put-object-retention

O código de exemplo a seguir mostra como usar `put-object-retention`.

AWS CLI

Como definir uma configuração de retenção de objetos para um objeto

O exemplo de `put-object-retention` a seguir define uma configuração de retenção de objetos para o objeto especificado até 2025-01-01.

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutObjectRetention](#) na Referência de AWS CLI Comandos.

put-public-access-block

O código de exemplo a seguir mostra como usar `put-public-access-block`.

AWS CLI

Para definir a configuração do bloco de acesso público para um bucket

O `put-public-access-block` exemplo a seguir define uma configuração restritiva de acesso público de blocos para o bucket especificado.

```
aws s3api put-public-access-block \
  --bucket my-bucket \
  --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutPublicAccessBlock](#) na Referência de AWS CLI Comandos.

register-transit-gateway

O código de exemplo a seguir mostra como usar `register-transit-gateway`.

AWS CLI

Para registrar um gateway de trânsito em uma rede global

O `register-transit-gateway` exemplo a seguir registra o gateway de trânsito `tgw-123abc05e04123abc` na rede global especificada.

```
aws networkmanager register-transit-gateway \
  --global-network-id global-network-01231231231231231 \
  --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc \
  --region us-west-2
```

Saída:

```
{
  "TransitGatewayRegistration": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
    "State": {
      "Code": "PENDING"
    }
  }
}
```

Para obter mais informações, consulte [Registros do Transit Gateway](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [RegisterTransitGateway](#) na Referência de AWS CLI Comandos.

reject-attachment

O código de exemplo a seguir mostra como usar reject-attachment.

AWS CLI

Para rejeitar um anexo

O reject-attachment exemplo a seguir rejeita uma solicitação de VPC anexo.

```
aws networkmanager reject-attachment \  
  --attachment-id attachment-03b7ea450134787da
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Aceitação de anexos](#) no Cloud WAN User Guide.

- Para API obter detalhes, consulte [RejectAttachment](#) na Referência de AWS CLI Comandos.

start-route-analysis

O código de exemplo a seguir mostra como usar start-route-analysis.

AWS CLI

Para iniciar a análise da rota

O `start-route-analysis` exemplo a seguir inicia a análise entre a origem e o destino, incluindo o opcional `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
```

Saída:

```
{  
  "RouteAnalysis": {  
    "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000"  
    "OwnerId": "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  
  }  
}
```

Para obter mais informações, consulte o [Route Analyzer](#) no Guia do usuário de redes AWS globais para gateways de trânsito.

- Para API obter detalhes, consulte [StartRouteAnalysis](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para aplicar tags a um recurso

O tag-resource exemplo a seguir aplica a tag Network=Northeast ao 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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Como remover as tags de um recurso

O untag-resource exemplo a seguir remove a tag com a chave Network do 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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-device

O código de exemplo a seguir mostra como usar update-device.

AWS CLI

Para atualizar um dispositivo

O update-device exemplo a seguir atualiza o dispositivo device-07f6fd08867abc123 especificando uma ID do site para o dispositivo.

```
aws networkmanager update-device \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --site-id site-444555aaabbb11223 \  
  --region us-west-2
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com dispositivos](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [UpdateDevice](#) na Referência de AWS CLI Comandos.

update-global-network

O código de exemplo a seguir mostra como usar `update-global-network`.

AWS CLI

Para atualizar uma rede global

O `update-global-network` exemplo a seguir atualiza a descrição da rede `global-network-01231231231231231`.

```
aws networkmanager update-global-network \
  --global-network-id global-network-01231231231231231 \
  --description "Head offices" \
  --region us-west-2
```

Saída:

```
{
  "GlobalNetwork": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/global-network-01231231231231231",
    "Description": "Head offices",
    "CreatedAt": 1575553525.0,
    "State": "UPDATING"
  }
}
```

Para obter mais informações, consulte [Redes globais](#) no Guia do Gerenciador de Redes do Transit Gateway.

- Para API obter detalhes, consulte [UpdateGlobalNetwork](#) na Referência de AWS CLI Comandos.

update-link

O código de exemplo a seguir mostra como usar `update-link`.

AWS CLI

Para atualizar um link

O `update-link` exemplo a seguir atualiza as informações de largura de banda do link `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
```

Saída:

```
{  
  "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": "UPDATING"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com links](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [UpdateLink](#) na Referência de AWS CLI Comandos.

update-site

O código de exemplo a seguir mostra como usar `update-site`.

AWS CLI

Para atualizar um site

O `update-site` exemplo a seguir atualiza a descrição do site `site-444555aaabbb11223` na rede global especificada.

```
aws networkmanager update-site \  
  --global-network-id global-network-01231231231231231 \  
  --site-id site-444555aaabbb11223 \  
  --description "New York Office site" \  
  --region us-west-2
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com sites](#) no Guia do Gerenciador de Rede do Transit Gateway.

- Para API obter detalhes, consulte [UpdateSite](#) na Referência de AWS CLI Comandos.

Exemplos do Nimble Studio usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Nimble Studio.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-eula

O código de exemplo a seguir mostra como usar `get-eula`.

AWS CLI

Para obter informações sobre seu estúdio

O `get-eula` exemplo a seguir lista as informações sobre um EULA.

```
aws nimble get-eula \  
  --eula-id "EULAid"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Aceite o EULA](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [GetEula](#) na Referência de AWS CLI Comandos.

get-launch-profile-details

O código de exemplo a seguir mostra como usar `get-launch-profile-details`.

AWS CLI

Para listar os widgets disponíveis

O `get-launch-profile-details` exemplo a seguir lista os detalhes sobre um perfil de lançamento.

```
aws nimble get-launch-profile-details \
  --studio-id "StudioID" \
  --launch-profile-id "LaunchProfileID"
```

Saída:

```
{
  "launchProfile": {
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q",
    "createdAt": "2022-01-27T21:18:59+00:00",
    "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
    "description": "The Launch Profile for the Render workers created by
StudioBuilder.",
    "ec2SubnetIds": [
      "subnet-EXAMPLE11111"
    ],
    "launchProfileId": "yeG7lDwNQEiwNTRT7DrV7Q",
    "launchProfileProtocolVersions": [
      "2021-03-31"
    ],
    "name": "RenderWorker-Default",
    "state": "READY",
    "statusCode": "LAUNCH_PROFILE_CREATED",
    "statusMessage": "Launch Profile has been created",
    "streamConfiguration": {
      "clipboardMode": "ENABLED",
      "ec2InstanceTypes": [
        "g4dn.4xlarge",
        "g4dn.8xlarge"
      ],
      "maxSessionLengthInMinutes": 690,
    }
  }
}
```

```
    "maxStoppedSessionLengthInMinutes": 0,
    "streamingImageIds": [
      "Cw_jXnp1QcSSXhE2hkNRoQ",
      "YGXAqgoWTnCNSV8VP20sHQ"
    ]
  },
  "studioComponentIds": [
    "_hR_-RaAReS0jAnLakbX7Q",
    "vQ5w_TbIRayPkAZgcbYRA",
    "ZQuMxN99Qfa_Js6ma9TwdA",
    "45Kj0SPPrzK20yvpCuQ6qw"
  ],
  "tags": {
    "resourceArn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG71DwNQEiwNTRT7DrV7Q"
  },
  "updatedAt": "2022-01-27T21:19:13+00:00",
  "updatedBy": "AR0A3002NEHCCYRNDIFT: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_TbIRayPkAZgcbYRA",
      "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"
    }
  ]
}

```

Para obter mais informações, consulte [Criação de perfis de lançamento](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [GetLaunchProfileDetails](#) na Referência de AWS CLI Comandos.

get-launch-profile

O código de exemplo a seguir mostra como usar `get-launch-profile`.

AWS CLI

Para listar os widgets disponíveis

O `get-launch-profile` exemplo a seguir lista informações sobre um perfil de lançamento.

```
aws nimble get-launch-profile \  
  --studio-id "StudioID" \  
  --launch-profile-id "LaunchProfileID"
```

Saída:

```
{  
  "launchProfile": {  
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/  
yeG7lDwNQEiwNTRT7DrV7Q",  
    "createdAt": "2022-01-27T21:18:59+00:00",  
    "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",  
    "description": "The Launch Profile for the Render workers created by  
StudioBuilder.",  
    "ec2SubnetIds": [  
      "subnet-EXAMPLE11111"  
    ],  
    "launchProfileId": "yeG7lDwNQEiwNTRT7DrV7Q",  
    "launchProfileProtocolVersions": [  
      "2021-03-31"  
    ],  
    "name": "RenderWorker-Default",  
    "state": "READY",  
    "statusCode": "LAUNCH_PROFILE_CREATED",  
    "statusMessage": "Launch Profile has been created",  
    "streamConfiguration": {  
      "clipboardMode": "ENABLED",  
      "ec2InstanceTypes": [  
        "g4dn.4xlarge",  
        "g4dn.8xlarge"  
      ],  
      "maxSessionLengthInMinutes": 690,  
      "maxStoppedSessionLengthInMinutes": 0,  
      "streamingImageIds": [  

```

```
        "Cw_jXnp1QcSSXhE2hkNRoQ",
        "YGXAqgoWTnCNSV8VP20sHQ"
    ]
},
"studioComponentIds": [
    "_hR_-RaAReS0jAnLakbX7Q",
    "vQ5w_TbIRayPkAZgcbyYRA",
    "ZQuMxN99Qfa_Js6ma9TwdA",
    "45Kj0SPPRzK20yvpCuQ6qw"
],
"tags": {},
"updatedAt": "2022-01-27T21:19:13+00:00",
"updatedBy": "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"
    }
]
}
}
```

Para obter mais informações, consulte [Criação de perfis de lançamento](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [GetLaunchProfile](#) na Referência de AWS CLI Comandos.

get-studio

O código de exemplo a seguir mostra como usar `get-studio`.

AWS CLI

Para obter informações sobre seu estúdio

O `get-studio` exemplo a seguir lista os estúdios em sua AWS conta.

```
aws nimble get-studio \  
  --studio-id "StudioID"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [O que é o Amazon Nimble Studio?](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [GetStudio](#) na Referência de AWS CLI Comandos.

list-eula-acceptances

O código de exemplo a seguir mostra como usar `list-eula-acceptances`.

AWS CLI

Para listar os widgets disponíveis

O `list-eula-acceptances` exemplo a seguir lista os aceitos EULAs em sua AWS conta.

```
aws nimble list-eula-acceptances \  
  --studio-id "StudioID"
```

Saída:

```
{  
  "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": "Rl-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": "wtp85BcSTa2NZeNRnMKdjw"
}
]
```

Para obter mais informações, consulte [Aceite o EULA](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [ListEulaAcceptances](#) na Referência de AWS CLI Comandos.

list-eulas

O código de exemplo a seguir mostra como usar `list-eulas`.

AWS CLI

Para listar os widgets disponíveis

O `list-eulas` exemplo a seguir lista os EULAs em sua AWS conta.

```
aws nimble list-eulas
```

Saída:

```
{
  "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-J0fM5S12hyIiwWIV6hw",
      "name": "Videolan-VLC",
      "updatedAt": "2021-04-20T16:45:24+00:00"
    },
    {
      "content": "https://code.visualstudio.com/license",
      "createdAt": "2021-04-20T16:45:23+00:00",
      "eulaId": "ggK2eIw6RQyt8PIee0lD3g",
      "name": "Microsoft-VSCode",
      "updatedAt": "2021-04-20T16:45:23+00:00"
    },
    {
      "content": "https://darbyjohnston.github.io/DJV/legal.html#License",
      "createdAt": "2021-04-20T16:45:23+00:00",
      "eulaId": "wtp85BcSTa2NZeNRnMKdjwt",
      "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"
    }
  ]
}
```

Para obter mais informações, consulte [Aceite o EULA](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [ListEulas](#) na Referência de AWS CLI Comandos.

list-launch-profiles

O código de exemplo a seguir mostra como usar `list-launch-profiles`.

AWS CLI

Para listar os widgets disponíveis

O `list-launch-profiles` exemplo a seguir lista os perfis de lançamento em sua AWS conta.

```
aws nimble list-launch-profiles \
  --studio-id "StudioID"
```


Saída:

```
{
  "launchProfiles": [
    {
      "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG71DwNQEiwNTRT7DrV7Q",
      "createdAt": "2022-01-27T21:18:59+00:00",
      "createdBy": "AROA3002NEHCCYRNDIFT: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_TbIRayPkAZgcbYRA",
        "ZQuMxN99Qfa_Js6ma9TwdA",
        "45Kj0SPPRzK20yvpCuQ6qw"
      ],
      "tags": {},
      "updatedAt": "2022-01-27T21:19:13+00:00",
      "updatedBy": "AROA3002NEHCCYRNDIFT: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": "ARO3002NEHCCYRNDIIFT: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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Criação de perfis de lançamento](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [ListLaunchProfiles](#) na Referência de AWS CLI Comandos.

list-studio-components

O código de exemplo a seguir mostra como usar `list-studio-components`.

AWS CLI

Para listar os widgets disponíveis

O `list-studio-components` exemplo a seguir lista os componentes do estúdio em sua AWS conta.

```
aws nimble list-studio-components \
  --studio-id "StudioID"
```

Saída:

```

{
  "studioComponents": [
    {
      "arn": "arn:aws:nimble:us-west-2:123456789012:studio-component/
ZQuMxN99Qfa_Js6ma9TwdA",
      "configuration": {
        "sharedFileSystemConfiguration": {
          "fileSystemId": "fs-EXAMPLE11111",
          "linuxMountPoint": "/mnt/fsxshare",
          "shareName": "share",
          "windowsMountDrive": "Z"
        }
      }
    }
  ]
}

```

```

    },
    "createdAt": "2022-01-27T21:15:34+00:00",
    "createdBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111",
    "description": "FSx for Windows",
    "ec2SecurityGroupIds": [
      "sg-EXAMPLE11111"
    ],
    "name": "FSxWindows",
    "state": "READY",
    "statusCode": "STUDIO_COMPONENT_CREATED",
    "statusMessage": "Studio Component has been created",
    "studioComponentId": "ZQuMxN99Qfa_Js6ma9TwdA",
    "subtype": "AMAZON_FSX_FOR_WINDOWS",
    "tags": {},
    "type": "SHARED_FILE_SYSTEM",
    "updatedAt": "2022-01-27T21:15:35+00:00",
    "updatedBy": "AROA3002NEHCCYRNDDIFT:i-EXAMPLE11111"
  },
  ...
}

```

Para obter mais informações, consulte [Como StudioBuilder funciona com o Amazon Nimble Studio no Guia](#) do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [ListStudioComponents](#) na Referência de AWS CLI Comandos.

list-studio-members

O código de exemplo a seguir mostra como usar `list-studio-members`.

AWS CLI

Para listar os widgets disponíveis

O `list-studio-members` exemplo a seguir lista os membros do estúdio disponíveis em sua AWS conta.

```
aws nimble list-studio-members \
  --studio-id "StudioID"
```

Saída:

```
{
  "members": [
    {
      "identityStoreId": "d-EXAMPLE11111",
      "persona": "ADMINISTRATOR",
      "principalId": "EXAMPLE11111-e9fd012a-94ad-4f16-9866-c69a63ab6486"
    }
  ]
}
```

Para obter mais informações, consulte [Adicionar usuários do estúdio](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [ListStudioMembers](#) na Referência de AWS CLI Comandos.

list-studios

O código de exemplo a seguir mostra como usar `list-studios`.

AWS CLI

Para listar seus estúdios

O `list-studios` exemplo a seguir lista os estúdios em sua AWS conta.

```
aws nimble list-studios
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [O que é o Amazon Nimble Studio?](#) no Guia do usuário do Amazon Nimble Studio.

- Para API obter detalhes, consulte [ListStudios](#) na Referência de AWS CLI Comandos.

OpenSearch Exemplos de serviços usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with OpenSearch Service.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-elasticsearch-domain

O código de exemplo a seguir mostra como usar `create-elasticsearch-domain`.

AWS CLI

Para criar um domínio do Amazon Elasticsearch Service

O `create-elasticsearch-domain` comando a seguir cria um novo domínio do Amazon Elasticsearch Service em um VPC e restringe o acesso a um único usuário. O Amazon ES infere o VPC ID da sub-rede e do grupo de segurança especificados. 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
```

Saída:

```
{  
  "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": "{\n\"Version\": \"2012-10-17\", \"Statement\": [\n{\n\"Effect\":\n\n\"Allow\", \"Principal\": {\n\"AWS\": \"arn:aws:iam::123456789012:root\"}, \"Action\":\n\n\"es:*\", \"Resource\": \"arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*\n\n\"}]]\",
    "AdvancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    },
    "EncryptionAtRestOptions": {
      "Enabled": false
    },
    "ARN": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example"
  }
}

```

Para obter mais informações, consulte [Criação e gerenciamento de domínios do Amazon Elasticsearch Service](#) no Guia do desenvolvedor do Amazon Elasticsearch Service.

- Para API obter detalhes, consulte [CreateElasticsearchDomain](#) na Referência de AWS CLI Comandos.

describe-elasticsearch-domain-config

O código de exemplo a seguir mostra como usar `describe-elasticsearch-domain-config`.

AWS CLI

Para obter detalhes da configuração do domínio

O `describe-elasticsearch-domain-config` exemplo a seguir fornece detalhes de configuração para um determinado domínio, juntamente com informações de status para cada componente individual do domínio.

```
aws es describe-elasticsearch-domain-config \  
--domain-name cli-example
```

Saída:

```
{  
  "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
    }
}
}
```

Para obter mais informações, consulte [Criação e gerenciamento de domínios do Amazon Elasticsearch Service](#) no Guia do desenvolvedor do Amazon Elasticsearch Service.

- Para API obter detalhes, consulte [DescribeElasticsearchDomainConfig](#) na Referência de AWS CLI Comandos.

describe-elasticsearch-domain

O código de exemplo a seguir mostra como usar describe-elasticsearch-domain.

AWS CLI

Para obter detalhes de um único domínio

O describe-elasticsearch-domain exemplo a seguir fornece detalhes de configuração para um determinado domínio.

```
aws es describe-elasticsearch-domain \  
  --domain-name cli-example
```

Saída:

```
{  
  "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
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Criação e gerenciamento de domínios do Amazon Elasticsearch Service](#) no Guia do desenvolvedor do Amazon Elasticsearch Service.

- Para API obter detalhes, consulte [DescribeElasticsearchDomain](#) na Referência de AWS CLI Comandos.

describe-elasticsearch-domains

O código de exemplo a seguir mostra como usar `describe-elasticsearch-domains`.

AWS CLI

Para obter detalhes de um ou mais domínios

O `describe-elasticsearch-domains` exemplo a seguir fornece detalhes de configuração para um ou mais domínios.

```
aws es describe-elasticsearch-domains \  
  --domain-names cli-example-1 cli-example-2
```

Saída:

```
{  
  "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
        }
    }
]
}

```

Para obter mais informações, consulte [Criação e gerenciamento de domínios do Amazon Elasticsearch Service](#) no Guia do desenvolvedor do Amazon Elasticsearch Service.

- Para API obter detalhes, consulte [DescribeElasticsearchDomains](#) na Referência de AWS CLI Comandos.

describe-reserved-elasticsearch-instances

O código de exemplo a seguir mostra como usar `describe-reserved-elasticsearch-instances`.

AWS CLI

Para ver todas as instâncias reservadas

O `describe-elasticsearch-domains` exemplo a seguir fornece um resumo de todas as instâncias que você reservou em uma região.

```
aws es describe-reserved-elasticsearch-instances
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Instâncias reservadas](#) no Guia do desenvolvedor do Amazon Elasticsearch Service.

- Para API obter detalhes, consulte [DescribeReservedElasticsearchInstances](#) na Referência de AWS CLI Comandos.

list-domain-names

O código de exemplo a seguir mostra como usar `list-domain-names`.

AWS CLI

Para listar todos os domínios

O `list-domain-names` exemplo a seguir fornece um resumo rápido de todos os domínios na região.

```
aws es list-domain-names
```

Saída:

```
{
  "DomainNames": [{
    "DomainName": "cli-example-1"
  },
  {
    "DomainName": "cli-example-2"
  }
]
}
```

Para obter mais informações, consulte [Criação e gerenciamento de domínios do Amazon Elasticsearch Service](#) no Guia do desenvolvedor do Amazon Elasticsearch Service.

- Para API obter detalhes, consulte [ListDomainNames](#) na Referência de AWS CLI Comandos.

AWS OpsWorks exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS OpsWorks.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

assign-instance

O código de exemplo a seguir mostra como usar `assign-instance`.

AWS CLI

Para atribuir uma instância registrada a uma camada

O exemplo a seguir atribui uma instância registrada a uma camada personalizada.

```
aws opsworks --region us-east-1 assign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --layer-ids 26cf1d32-6876-42fa-bbf1-9cad0bfff938
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Atribuição de uma instância registrada a uma camada](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [AssignInstance](#) na Referência de AWS CLI Comandos.

assign-volume

O código de exemplo a seguir mostra como usar `assign-volume`.

AWS CLI

Para atribuir um volume registrado a uma instância

O exemplo a seguir atribui um volume registrado do Amazon Elastic Block Store (AmazonEBS) a uma instância. O volume é identificado por seu ID de volume, que é atribuído quando você registra o GUID volume em uma pilha, não pelo ID de volume do Amazon Elastic Compute Cloud (AmazonEC2). AWS OpsWorks Antes de executar `assign-volume`, você deve primeiro executar `update-volume` para atribuir um ponto de montagem ao volume.

```
aws opsworks --region us-east-1 assign-volume --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --volume-id 26cf1d32-6876-42fa-bbf1-9cad0bfff938
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Atribuição de EBS volumes da Amazon a uma instância](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [AssignVolume](#) na Referência de AWS CLI Comandos.

associate-elastic-ip

O código de exemplo a seguir mostra como usar `associate-elastic-ip`.

AWS CLI

Para associar um endereço IP elástico a uma instância

O exemplo a seguir associa um endereço IP elástico a uma instância especificada.

```
aws opsworks --region us-east-1 associate-elastic-ip --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f --elastic-ip 54.148.130.96
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Gerenciamento de recursos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [AssociateElasticIp](#) na Referência de AWS CLI Comandos.

attach-elastic-load-balancer

O código de exemplo a seguir mostra como usar `attach-elastic-load-balancer`.

AWS CLI

Para anexar um balanceador de carga a uma camada

O exemplo a seguir anexa um balanceador de carga, identificado por seu nome, a uma camada especificada.

```
aws opsworks --region us-east-1 attach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Elastic Load Balancing no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [AttachElasticLoadBalancer](#) na Referência de AWS CLI Comandos.

create-app

O código de exemplo a seguir mostra como usar `create-app`.

AWS CLI

Exemplo 1: Para criar um aplicativo

O exemplo a seguir cria um PHP aplicativo chamado SimplePHPApp a partir do código armazenado em um GitHub repositório. O comando usa a forma abreviada da definição da fonte do aplicativo.

```
aws opsworks create-app \  
  --region us-east-1 \  
  --stack-id f6673d70-32e6-4425-8999-265dd002fec7 \  
  --name SimplePHPApp \  
  --type php \  
  --app-source Type=git,Url=https://github.com/amazonwebservices/opsworks-demo-php-simple-app.git,Revision=version1
```

Saída:

```
{  
  "AppId": "6cf5163c-a951-444f-a8f7-3716be75f2a2"  
}
```

Exemplo 2: Para criar um aplicativo com um banco de dados anexado

O exemplo a seguir cria um JSP aplicativo a partir do código armazenado em um arquivo.zip em um bucket público do S3. Ele anexa uma RDS instância de banco de dados para servir como armazenamento de dados do aplicativo. As fontes do aplicativo e do banco de dados são definidas em JSON arquivos separados que estão no diretório a partir do qual você executa o comando.

```
aws opsworks create-app \  
  --region us-east-1 \  
  --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8 \  
  --app-source Type=s3,Url=s3://mybucket/myapp.zip,Revision=1 \  
  --db-source Type=rds,Url=rds://mydbinstance.amazonaws.com,Revision=1
```



```
--name SimpleJSP \  
--type java \  
--app-source file://appsource.json \  
--data-sources file://datasource.json
```

As informações da fonte do aplicativo estão em `appsource.json` e contêm o seguinte.

```
{  
  "Type": "archive",  
  "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"  
}
```

As informações da fonte do banco de dados `datasource.json` estão em e contêm o seguinte.

```
[  
  {  
    "Type": "RdsDbInstance",  
    "Arn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",  
    "DatabaseName": "mydb"  
  }  
]
```

Observação: para uma RDS instância de banco de dados, você deve primeiro usar `register-rds-db-instance` para registrar a instância com a pilha. Para instâncias do My SQL App Server, `Type` defina como `OpsworksMySQLInstance`. Essas instâncias são criadas por AWS OpsWorks, portanto, não precisam ser registradas.

Saída:

```
{  
  "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"  
}
```

Para obter mais informações, consulte [Adicionar aplicativos no Guia AWS OpsWorks do usuário](#).

- Para API obter detalhes, consulte [CreateApp](#) na Referência de AWS CLI Comandos.

create-deployment

O código de exemplo a seguir mostra como usar `create-deployment`.

AWS CLI

Exemplo 1: para implantar aplicativos e executar comandos de pilha

Os exemplos a seguir mostram como usar o `create-deployment` comando para implantar aplicativos e executar comandos de pilha. Observe que os caracteres quote (") no JSON objeto que especifica o comando são todos precedidos por caracteres de escape (\). Sem os caracteres de escape, o comando pode retornar um JSON erro inválido.

O `create-deployment` exemplo a seguir implanta um aplicativo em uma pilha especificada.

```
aws opsworks create-deployment \  
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \  
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \  
  --command "{\"Name\":\"deploy\"}"
```

Saída:

```
{  
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"  
}
```

Exemplo 2: Para implantar um aplicativo Rails e migrar o banco de dados

O `create-deployment` comando a seguir implanta um aplicativo Ruby on Rails em uma pilha especificada e migra o banco de dados.

```
aws opsworks create-deployment \  
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \  
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \  
  --command "{\"Name\":\"deploy\", \"Args\":{\"migrate\":[\"true\"]}]}"
```

Saída:

```
{  
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"  
}
```

Para obter mais informações sobre implantação, consulte [Implantação de aplicativos](#) no Guia do AWS OpsWorks usuário.

Exemplo 3: Executar uma receita

O `create-deployment` comando a seguir executa uma receita personalizada, `phpapp::appsetup`, nas instâncias em uma pilha especificada.

```
aws opsworks create-deployment \  
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \  
  --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":  
[\"phpapp::appsetup\"]}}"
```

Saída:

```
{  
  "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"  
}
```

Para obter mais informações, consulte [Executar comandos do Stack](#) no Guia do AWS OpsWorks usuário.

Exemplo 4: Instalar dependências

O `create-deployment` comando a seguir instala dependências, como pacotes ou gems Ruby, nas instâncias em uma pilha especificada.

```
aws opsworks create-deployment \  
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \  
  --command "{\"Name\":\"install_dependencies\"}"
```

Saída:

```
{  
  "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"  
}
```

Para obter mais informações, consulte [Executar comandos do Stack](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [CreateDeployment](#) na Referência de AWS CLI Comandos.

create-instance

O código de exemplo a seguir mostra como usar `create-instance`.

AWS CLI

Para criar uma instância

O `create-instance` comando a seguir cria uma instância `m1.large` Amazon Linux chamada `myinstance1` em uma pilha especificada. A instância é atribuída a uma camada.

```
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"
```

Para usar um nome gerado automaticamente, chame `get-hostname-suggestion`, que gera um nome de host com base no tema que você especificou ao criar a pilha. Em seguida, passe esse nome para o argumento `hostname`.

Saída:

```
{
  "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Mais informações

Para obter mais informações, consulte [Adicionar uma instância a uma camada](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [CreateInstance](#) na Referência de AWS CLI Comandos.

`create-layer`

O código de exemplo a seguir mostra como usar `create-layer`.

AWS CLI

Para criar uma camada

O `create-layer` comando a seguir cria uma camada do PHP App Server chamada `MyPHPLayer` em uma pilha especificada.

```
aws opsworks create-layer --region us-east-1 --stack-id f6673d70-32e6-4425-8999-265dd002fec7 --type php-app --name MyPHPLayer --shortname myphpLayer
```

Saída:

```
{
  "LayerId": "0b212672-6b4b-40e4-8a34-5a943cf2e07a"
}
```

Mais informações

Para obter mais informações, consulte [Como criar uma camada](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [CreateLayer](#) na Referência de AWS CLI Comandos.

create-server

O código de exemplo a seguir mostra como usar `create-server`.

AWS CLI

Para criar um servidor

O `create-server` exemplo a seguir cria um novo servidor Chef Automate chamado `automate-06` em sua região padrão. Observe que os padrões são usados para a maioria das outras configurações, como o número de backups a serem retidos e os horários de início da manutenção e do backup. Antes de executar um `create-server` comando, preencha os pré-requisitos em [Introdução ao Chef Automate no Guia do usuário do AWS Opsworks 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"
```

Saída:

```
{
  "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" ]
  }
}

```

Para obter mais informações, consulte [CreateServer](#) a API Referência do AWS OpsWorks Chef Automate.

- Para API obter detalhes, consulte [CreateServer](#) na Referência de AWS CLI Comandos.

create-stack

O código de exemplo a seguir mostra como usar create-stack.

AWS CLI

Para criar uma pilha

O create-stack comando a seguir cria uma pilha chamada 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
```

Os parâmetros `service-role-arn` e `default-instance-profile-arn` são obrigatórios. Você normalmente usa os que são AWS OpsWorks criados para você ao criar sua primeira pilha. Para obter os Amazon Resource Names (ARNs) para sua conta, acesse o IAM console, escolha Roles no painel de navegação, escolha a função ou o perfil e escolha a Summary guia.

Saída:

```
{  
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"  
}
```

Mais informações

Para obter mais informações, consulte Criar uma nova pilha no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [CreateStack](#) na Referência de AWS CLI Comandos.

create-user-profile

O código de exemplo a seguir mostra como usar `create-user-profile`.

AWS CLI

Para criar um perfil de usuário

Você importa um usuário do AWS Identity and Access Manager (IAM) AWS OpsWorks ligando `create-user-profile` para criar um perfil de usuário. O exemplo a seguir cria um perfil de usuário para o `cli-user-test` IAM usuário, que é identificado pelo Amazon Resource Name (ARN). O exemplo atribui ao usuário um SSH nome de usuário `myusername` e ativa o autogerenciamento, o que permite que o usuário especifique uma chave SSH pública.

```
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
```

Saída:

```
{
  "IamUserArn": "arn:aws:iam::123456789102:user/cli-user-test"
}
```

Dica: Esse comando importa um IAM usuário para AWS OpsWorks, mas somente com as permissões concedidas pelas políticas anexadas. Você pode conceder AWS OpsWorks permissões por pilha usando o `set-permissions` comando.

Mais informações

Para obter mais informações, consulte [Importação de usuários AWS OpsWorks no Guia do AWS OpsWorks usuário](#).

- Para API obter detalhes, consulte [CreateUserProfile](#) na Referência de AWS CLI Comandos.

delete-app

O código de exemplo a seguir mostra como usar `delete-app`.

AWS CLI

Para excluir um aplicativo

O exemplo a seguir exclui um aplicativo especificado, que é identificado pelo ID do aplicativo. Você pode obter uma ID do aplicativo acessando a página de detalhes do aplicativo no AWS OpsWorks console ou executando o `describe-apps` comando.

```
aws opsworks delete-app --region us-east-1 --app-id 577943b9-2ec1-4baf-  
a7bf-1d347601edc5
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Aplicativos no Guia AWS OpsWorks do usuário](#).

- Para API obter detalhes, consulte [DeleteApp](#) na Referência de AWS CLI Comandos.

delete-instance

O código de exemplo a seguir mostra como usar `delete-instance`.

AWS CLI

Para excluir uma instância

O `delete-instance` exemplo a seguir exclui uma instância especificada, que é identificada pelo ID da instância. Você pode encontrar um ID de instância abrindo a página de detalhes da instância no AWS OpsWorks console ou executando o `describe-instances` comando.

Se a instância estiver on-line, primeiro você deverá interrompê-la chamando `stop-instance`, em seguida, aguardar até que a instância seja interrompida. Execute `describe-instances` para verificar o status da instância.

Para remover os EBS volumes da Amazon ou os endereços IP elásticos da instância, adicione os `--delete-elastic-ip` argumentos `--delete-volumes` ou, respectivamente.

```
aws opsworks delete-instance \  
  --region us-east-1 \  
  --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluir AWS OpsWorks instâncias](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeleteInstance](#) na Referência de AWS CLI Comandos.

delete-layer

O código de exemplo a seguir mostra como usar `delete-layer`.

AWS CLI

Para excluir uma camada

O exemplo a seguir exclui uma camada especificada, que é identificada por sua ID de camada. Você pode obter um ID de camada acessando a página de detalhes da camada no AWS OpsWorks console ou executando o `describe-layers` comando.

Nota: Antes de excluir uma camada, você deve usar `delete-instance` para excluir todas as instâncias da camada.

```
aws opsworks delete-layer --region us-east-1 --layer-id a919454e-b816-4598-b29a-5796afb498ed
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Excluir AWS OpsWorks instâncias no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeleteLayer](#) na Referência de AWS CLI Comandos.

delete-stack

O código de exemplo a seguir mostra como usar `delete-stack`.

AWS CLI

Para excluir uma pilha

O exemplo a seguir exclui uma pilha especificada, que é identificada pelo ID da pilha. Você pode obter um ID da pilha clicando em Configurações da pilha no AWS OpsWorks console ou executando o `describe-stacks` comando.

Observação: antes de excluir uma camada, você deve usar `delete-appdelete-instance`, e `delete-layer` para excluir todos os aplicativos, instâncias e camadas da pilha.

```
aws opsworks delete-stack --region us-east-1 --stack-id 154a9d89-7e9e-433b-8de8-617e53756c84
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Desligar uma pilha no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeleteStack](#) na Referência de AWS CLI Comandos.

delete-user-profile

O código de exemplo a seguir mostra como usar `delete-user-profile`.

AWS CLI

Para excluir um perfil de usuário e remover um IAM usuário do AWS OpsWorks

O exemplo a seguir exclui o perfil de usuário de um usuário especificado do AWS Identity and Access Management (IAM), identificado pelo Amazon Resource Name (ARN). A operação remove o usuário AWS OpsWorks, mas não o IAM exclui. Você deve usar o IAM console, CLI, ou API para essa tarefa.

```
aws opsworks --region us-east-1 delete-user-profile --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Importação de usuários AWS OpsWorks no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeleteUserProfile](#) na Referência de AWS CLI Comandos.

deregister-elastic-ip

O código de exemplo a seguir mostra como usar deregister-elastic-ip.

AWS CLI

Para cancelar o registro de um endereço IP elástico de uma pilha

O exemplo a seguir cancela o registro de um endereço IP elástico, identificado por seu endereço IP, de sua pilha.

```
aws opsworks deregister-elastic-ip --region us-east-1 --elastic-ip 54.148.130.96
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Cancelamento do registro de endereços IP elásticos no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeregisterElasticIp](#) na Referência de AWS CLI Comandos.

deregister-instance

O código de exemplo a seguir mostra como usar `deregister-instance`.

AWS CLI

Para cancelar o registro de uma instância registrada de uma pilha

O `deregister-instance` comando a seguir cancela o registro de uma instância registrada de sua pilha.

```
aws opsworks --region us-east-1 deregister-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Cancelamento do registro de uma instância registrada no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeregisterInstance](#) na Referência de AWS CLI Comandos.

deregister-rds-db-instance

O código de exemplo a seguir mostra como usar `deregister-rds-db-instance`.

AWS CLI

Para cancelar o registro de uma instância de RDS banco de dados Amazon de uma pilha

O exemplo a seguir cancela o registro de uma RDS instância de banco de dados, identificada por `itsARN`, de sua pilha.

```
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:123456789012:db:clitestdb
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Cancelamento do registro de instâncias da RDS Amazon](#) no Guia ASW OpsWorks do usuário.

ID da instância: clitestdb Nome de usuário principal: cliuser Master: some23! PWD Nome do banco de dados pwd: mydb aws opsworks deregister-rds-db-instance --region us-east-1 --arn:aws:rds:us-west-2:645732743964:db:clitestdb rds-db-instance-arn

- Para API obter detalhes, consulte [DeregisterRdsDbInstance](#) na Referência de AWS CLI Comandos.

deregister-volume

O código de exemplo a seguir mostra como usar `deregister-volume`.

AWS CLI

Para cancelar o registro de um volume da Amazon EBS

O exemplo a seguir cancela o registro de um EBS volume de sua pilha. O volume é identificado pelo ID do volume, que é GUID aquele AWS OpsWorks atribuído quando você registrou o volume na pilha, não pelo ID do EC2 volume.

```
aws opsworks deregister-volume --region us-east-1 --volume-id 5c48ef52-3144-4bf5-beaa-fda4deb23d4d
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Cancelamento do registro de volumes da EBS Amazon](#) no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DeregisterVolumen](#) na Referência de AWS CLI Comandos.

describe-apps

O código de exemplo a seguir mostra como usar `describe-apps`.

AWS CLI

Para descrever aplicativos

O `describe-apps` comando a seguir descreve os aplicativos em uma pilha especificada.

```
aws opsworks describe-apps \  
  --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte Aplicativos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeApps](#) na Referência de AWS CLI Comandos.

describe-commands

O código de exemplo a seguir mostra como usar `describe-commands`.

AWS CLI

Para descrever comandos

O `describe-commands` comando a seguir descreve os comandos em uma instância especificada.

```
aws opsworks describe-commands \
  --instance-id 8c2673b9-3fe5-420d-9cfa-78d875ee7687 \
  --region us-east-1
```

Saída:

```
{
  "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
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte Eventos AWS OpsWorks do ciclo de vida no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeCommands](#) na Referência de AWS CLI Comandos.

describe-deployments

O código de exemplo a seguir mostra como usar `describe-deployments`.

AWS CLI

Para descrever implantações

O `describe-deployments` comando a seguir descreve as implantações em uma pilha especificada.

```
aws opsworks --region us-east-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

Saída:

```
{
  "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"
}
]
}
```

Mais informações

Para obter mais informações, consulte Implantação de aplicativos no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeDeployments](#) na Referência de AWS CLI Comandos.

describe-elastic-ips

O código de exemplo a seguir mostra como usar `describe-elastic-ips`.

AWS CLI

Para descrever instâncias de IP elástico

O `describe-elastic-ips` comando a seguir descreve os endereços IP elásticos em uma instância especificada.

```
aws opsworks --region us-east-1 describe-elastic-ips --instance-id b62f3e04-
e9eb-436c-a91f-d9e9a396b7b0
```

Saída:

```
{
  "ElasticIps": [
```

```
{
  "Ip": "192.0.2.0",
  "Domain": "standard",
  "Region": "us-west-2"
}
]
```

Mais informações

Para obter mais informações, consulte Instâncias no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeElasticIps](#) na Referência de AWS CLI Comandos.

describe-elastic-load-balancers

O código de exemplo a seguir mostra como usar `describe-elastic-load-balancers`.

AWS CLI

Para descrever os balanceadores de carga elásticos de uma pilha

O `describe-elastic-load-balancers` comando a seguir descreve os balanceadores de carga de uma pilha especificada.

```
aws opsworks --region us-west-2 describe-elastic-load-balancers --stack-id 6f4660e5-37a6-4e42-bfa0-1358ebd9c182
```

Saída: essa pilha específica tem um balanceador de carga.

```
{
  "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"  
  }  
]  
}
```

Mais informações

Para obter mais informações, consulte Aplicativos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeElasticLoadBalancers](#) na Referência de AWS CLI Comandos.

describe-instances

O código de exemplo a seguir mostra como usar `describe-instances`.

AWS CLI

Para descrever instâncias

O `describe-instances` comando a seguir descreve as instâncias em uma pilha especificada:

```
aws opsworks --region us-east-1 describe-instances --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Saída: O exemplo de saída a seguir é para uma pilha com duas instâncias. A primeira é uma EC2 instância registrada e a segunda foi criada por 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"
  }
]
}
```

Mais informações

Para obter mais informações, consulte Instâncias no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeInstances](#) na Referência de AWS CLI Comandos.

describe-layers

O código de exemplo a seguir mostra como usar `describe-layers`.

AWS CLI

Para descrever as camadas de uma pilha

O `describe-layers` comando a seguir descreve as camadas em uma pilha especificada:

```
aws opsworks --region us-east-1 describe-layers --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

Saída:

```
{
  "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": []
}
]
```

Mais informações

Para obter mais informações, consulte Camadas no Guia AWS OpsWorks do Usuário.

- Para API obter detalhes, consulte [DescribeLayers](#) na Referência de AWS CLI Comandos.

describe-load-based-auto-scaling

O código de exemplo a seguir mostra como usar `describe-load-based-auto-scaling`.

AWS CLI

Para descrever a configuração de escalabilidade baseada em carga de uma camada

O exemplo a seguir descreve a configuração de escalabilidade baseada em carga de uma camada especificada. A camada é identificada por seu ID de camada, que você pode encontrar na página de detalhes da camada ou executando `describe-layers`.

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

Saída: A camada de exemplo tem uma única instância baseada em carga.

```
{
  "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"
    }
  ]
}
```

Mais informações

Para obter mais informações, consulte [Como funciona a escalabilidade automática baseada em carga](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeLoadBasedAutoScaling](#) na Referência de AWS CLI Comandos.

describe-my-user-profile

O código de exemplo a seguir mostra como usar `describe-my-user-profile`.

AWS CLI

Para obter o perfil de um usuário

O exemplo a seguir mostra como obter o perfil do usuário AWS Identity and Access Management (IAM) que está executando o comando.

```
aws opsworks --region us-east-1 describe-my-user-profile
```

Resultado: Para resumir, a maior parte da chave SSH pública do usuário é substituída por uma elipse (...).

```
{
  "UserProfile": {
    "IamUserArn": "arn:aws:iam::123456789012:user/myusername",
    "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQ...3LQ4aX9jpxQw== rsa-
key-20141104",
    "Name": "myusername",
    "SshUsername": "myusername"
  }
}
```

Mais informações

Para obter mais informações, consulte [Importação de usuários AWS OpsWorks](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeMyUserProfile](#) na Referência de AWS CLI Comandos.

describe-permissions

O código de exemplo a seguir mostra como usar `describe-permissions`.

AWS CLI

Para obter o nível de permissão por pilha de AWS OpsWorks um usuário

O exemplo a seguir mostra como obter o nível de permissão de um usuário do AWS Identity and Access Management (IAM) em uma pilha especificada.

```
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
```

Saída:

```
{
  "Permissions": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "Level": "manage",
      "AllowSudo": true,
      "AllowSsh": true
    }
  ]
}
```

Mais informações

Para obter mais informações, consulte [Conceder níveis de permissões por pilha](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribePermissions](#) na Referência de AWS CLI Comandos.

describe-raid-arrays

O código de exemplo a seguir mostra como usar `describe-raid-arrays`.

AWS CLI

Para descrever RAID matrizes

O exemplo a seguir descreve as RAID matrizes anexadas às instâncias em uma pilha especificada.

```
aws opsworks --region us-east-1 describe-raid-arrays --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

Saída: A seguir está a saída para uma pilha com uma RAID matriz.

```
{
```

```
"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
  }
]
```

Para obter mais informações, consulte EBS Volumes no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeRaidArrays](#) na Referência de AWS CLI Comandos.

describe-rds-db-instances

O código de exemplo a seguir mostra como usar `describe-rds-db-instances`.

AWS CLI

Para descrever as instâncias Amazon RDS registradas de uma pilha

O exemplo a seguir descreve as RDS instâncias da Amazon registradas com uma pilha especificada.

```
aws opsworks --region us-east-1 describe-rds-db-instances --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

Saída: a seguir está a saída de uma pilha com uma RDS instância registrada.

```
{
  "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"
}
]
}
```

Para obter mais informações, consulte Gerenciamento de recursos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeRdsDbInstances](#) na Referência de AWS CLI Comandos.

describe-stack-provisioning-parameters

O código de exemplo a seguir mostra como usar `describe-stack-provisioning-parameters`.

AWS CLI

Para retornar os parâmetros de provisionamento de uma pilha

O `describe-stack-provisioning-parameters` exemplo a seguir retorna os parâmetros de provisionamento de uma pilha especificada. Os parâmetros de provisionamento incluem configurações como o local de instalação do agente e a chave pública OpsWorks usada para gerenciar o agente em instâncias em uma pilha.

```
aws opsworks describe-stack-provisioning-parameters \
  --stack-id 62744d97-6faf-4ecb-969b-a086fEXAMPLE
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Executar comandos do Stack](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeStackProvisioningParameters](#) na Referência de AWS CLI Comandos.

describe-stack-summary

O código de exemplo a seguir mostra como usar `describe-stack-summary`.

AWS CLI

Para descrever a configuração de uma pilha

O `describe-stack-summary` comando a seguir retorna um resumo da configuração da pilha especificada.

```
aws opsworks --region us-east-1 describe-stack-summary --stack-id 8c428b08-
a1a1-46ce-a5f8-feddc43771b8
```

Saída:

```
{
```

```
"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/"
}
```

Mais informações

Para obter mais informações, consulte Stacks no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeStackSummary](#) na Referência de AWS CLI Comandos.

describe-stacks

O código de exemplo a seguir mostra como usar describe-stacks.

AWS CLI

Para descrever pilhas

O describe-stacks comando a seguir descreve as pilhas de uma conta.

```
aws opsworks --region us-east-1 describe-stacks
```

Saída:

```
{
  "Stacks": [
    {
      "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
      "StackId": "aeb7523e-7c8b-49d4-b866-03aae9d4fbcb",
      "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"
}
]
```

```
}
```

Mais informações

Para obter mais informações, consulte Stacks no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeStacks](#) na Referência de AWS CLI Comandos.

describe-timebased-auto-scaling

O código de exemplo a seguir mostra como usar `describe-timebased-auto-scaling`.

AWS CLI

Para descrever a configuração de escalabilidade baseada em tempo de uma instância

O exemplo a seguir descreve a configuração de escalabilidade baseada em tempo de uma instância especificada. A instância é identificada pelo ID da instância, que você pode encontrar na página de detalhes das instâncias ou na execução de `describe-instances`.

```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

Saída: o exemplo tem uma única instância baseada em 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"
        }
      }
    }
  ]
}
```

```
    }
  }
}
]
```

Mais informações

Para obter mais informações, consulte [Como funciona a escalabilidade automática com base no tempo](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeTimebasedAutoScaling](#) na Referência de AWS CLI Comandos.

describe-user-profiles

O código de exemplo a seguir mostra como usar `describe-user-profiles`.

AWS CLI

Para descrever perfis de usuário

O `describe-user-profiles` comando a seguir descreve os perfis de usuário da conta.

```
aws opsworks --region us-east-1 describe-user-profiles
```

Saída:

```
{
  "UserProfiles": [
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/someuser",
      "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEakOuP7i80q3Cko...",
      "AllowSelfManagement": true,
      "Name": "someuser",
      "SshUsername": "someuser"
    },
    {
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "AllowSelfManagement": true,
      "Name": "cli-user-test",

```

```
    "SshUsername": "myusername"
  }
]
}
```

Mais informações

Para obter mais informações, consulte Gerenciamento de AWS OpsWorks usuários no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeUserProfiles](#) na Referência de AWS CLI Comandos.

describe-volumes

O código de exemplo a seguir mostra como usar `describe-volumes`.

AWS CLI

Para descrever os volumes de uma pilha

O exemplo a seguir descreve os EBS volumes de uma pilha.

```
aws opsworks --region us-east-1 describe-volumes --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Saída:

```
{
  "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
    }
  ]
}
```

```
]
}
```

Mais informações

Para obter mais informações, consulte Gerenciamento de recursos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeVolumes](#) na Referência de AWS CLI Comandos.

detach-elastic-load-balancer

O código de exemplo a seguir mostra como usar `detach-elastic-load-balancer`.

AWS CLI

Para separar um balanceador de carga de sua camada

O exemplo a seguir separa um balanceador de carga, identificado por seu nome, de sua camada.

```
aws opsworks --region us-east-1 detach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Elastic Load Balancing no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DetachElasticLoadBalancer](#) na Referência de AWS CLI Comandos.

disassociate-elastic-ip

O código de exemplo a seguir mostra como usar `disassociate-elastic-ip`.

AWS CLI

Para desassociar um endereço IP elástico de uma instância

O exemplo a seguir desassocia um endereço IP elástico de uma instância especificada.

```
aws opsworks --region us-east-1 disassociate-elastic-ip --elastic-ip 54.148.130.96
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Gerenciamento de recursos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DisassociateElasticIp](#) na Referência de AWS CLI Comandos.

get-hostname-suggestion

O código de exemplo a seguir mostra como usar `get-hostname-suggestion`.

AWS CLI

Para obter o próximo nome de host para uma camada

O exemplo a seguir obtém o próximo nome de host gerado para uma camada especificada. A camada usada neste exemplo é uma camada do Java Application Server com uma instância. O tema do nome de host da pilha é o padrão, `Layer_Dependent`.

```
aws opsworks --region us-east-1 get-hostname-suggestion --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Saída:

```
{
  "Hostname": "java-app2",
  "LayerId": "888c5645-09a5-4d0e-95a8-812ef1db76a4"
}
```

Mais informações

Para obter mais informações, consulte Criar uma nova pilha no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [GetHostnameSuggestion](#) na Referência de AWS CLI Comandos.

reboot-instance

O código de exemplo a seguir mostra como usar `reboot-instance`.

AWS CLI

Para reinicializar uma instância

O exemplo a seguir reinicializa uma instância.

```
aws opsworks --region us-east-1 reboot-instance --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Reinicializando uma instância no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [RebootInstance](#) na Referência de AWS CLI Comandos.

register-elastic-ip

O código de exemplo a seguir mostra como usar `register-elastic-ip`.

AWS CLI

Para registrar um endereço IP elástico com uma pilha

O exemplo a seguir registra um endereço IP elástico, identificado por seu endereço IP, com uma pilha especificada.

Observação: o endereço IP elástico deve estar na mesma região da pilha.

```
aws opsworks register-elastic-ip --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --elastic-ip 54.148.130.96
```

Saída

```
{
```

```
"ElasticIp": "54.148.130.96"  
}
```

Mais informações

Para obter mais informações, consulte Registro de endereços IP elásticos com uma pilha no Guia do OpsWorks usuário.

- Para API obter detalhes, consulte [RegisterElasticIp](#) na Referência de AWS CLI Comandos.

register-rds-db-instance

O código de exemplo a seguir mostra como usar `register-rds-db-instance`.

AWS CLI

Para registrar uma RDS instância da Amazon com uma pilha

O exemplo a seguir registra uma RDS instância de banco de dados Amazon, identificada por seu Amazon Resource Name (ARN), com uma pilha especificada. Ela também especifica o nome de usuário e a senha principais da instância. Observe que AWS OpsWorks isso não valida nenhum desses valores. Se algum deles estiver incorreto, seu aplicativo não conseguirá se conectar ao banco de dados.

```
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
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Registro de RDS instâncias da Amazon com uma pilha no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [RegisterRdsDbInstance](#) na Referência de AWS CLI Comandos.

register-volume

O código de exemplo a seguir mostra como usar `register-volume`.

AWS CLI

Para registrar um EBS volume da Amazon com uma pilha

O exemplo a seguir registra um EBS volume da Amazon, identificado por seu ID de volume, com uma pilha especificada.

```
aws opsworks register-volume --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --ec-2-volume-id vol-295c1638
```

Saída:

```
{
  "VolumeId": "ee08039c-7cb7-469f-be10-40fb7f0c05e8"
}
```

Mais informações

Para obter mais informações, consulte Registrando EBS volumes da Amazon com uma pilha no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [RegisterVolume](#) na Referência de AWS CLI Comandos.

register

O código de exemplo a seguir mostra como usar `register`.

AWS CLI

Para registrar instâncias com uma pilha

Os exemplos a seguir mostram várias maneiras de registrar instâncias com uma pilha criada fora do AWS Opsworks. Você pode executar a `register` partir da instância a ser registrada ou de uma estação de trabalho separada. Para obter mais informações, consulte Registro de instâncias locais EC2 e da Amazon no Guia do AWS OpsWorks usuário.

Nota: Para resumir, os exemplos omitem o `region` argumento.

Para registrar uma EC2 instância da Amazon

Para indicar que você está registrando uma EC2 instância, defina o `--infrastructure-class` argumento como. `ec2`

O exemplo a seguir registra uma EC2 instância com a pilha especificada de uma estação de trabalho separada. A instância é identificada por seu EC2 ID, `i-12345678`. O exemplo usa o nome de SSH usuário padrão da estação de trabalho e tenta fazer login na instância usando técnicas de autenticação que não exigem uma senha, como uma SSH chave privada padrão. Se isso falhar, `register` consulta a senha.

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb i-12345678
```

O exemplo a seguir registra uma EC2 instância com a pilha especificada de uma estação de trabalho separada. Ele usa os `--ssh-private-key` argumentos `--ssh-username` e para especificar explicitamente o SSH nome de usuário e o arquivo de chave privada que o comando usa para fazer login na instância. `ec2-user` é o nome de usuário padrão para instâncias do Amazon Linux. Use `ubuntu` para instâncias do 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
```

O exemplo a seguir registra a EC2 instância que está executando o `register` comando. Faça login na instância com SSH e execute `register` com o `--local` argumento em vez de um ID de instância ou nome de host.

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

Para registrar uma instância local

Para indicar que você está registrando uma instância local, defina o `--infrastructure-class` argumento como `on-premises`

O exemplo a seguir registra uma instância local existente com uma pilha especificada de uma estação de trabalho separada. A instância é identificada por seu endereço IP, `192.0.2.3`. O exemplo usa o nome de SSH usuário padrão da estação de trabalho e tenta fazer login na instância usando técnicas de autenticação que não exigem uma senha, como uma SSH chave privada padrão. Se isso falhar, `register` consulta a senha.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

O exemplo a seguir registra uma instância local com uma pilha especificada de uma estação de trabalho separada. A instância é identificada por seu nome de host, `host1`. Os `--override-...` argumentos são direcionados AWS OpsWorks para exibição `webserver1` como nome do host `192.0.2.3` e `10.0.0.2` como endereços IP públicos e privados da instância, respectivamente.

```
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
```

O exemplo a seguir registra uma instância local com uma pilha especificada de uma estação de trabalho separada. A instância é identificada por seu endereço IP. `register` faz login na instância usando o nome de SSH usuário e o arquivo de chave privada especificados.

```
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
```

O exemplo a seguir registra uma instância local existente com uma pilha especificada de uma estação de trabalho separada. O comando faz login na instância usando uma string de SSH comando personalizada que especifica a SSH senha e o endereço IP da instância.

```
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"
```

O exemplo a seguir registra a instância local que está executando o `register` comando. Faça login na instância com SSH e execute `register` com o `--local` argumento em vez de um ID de instância ou nome de host.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

Saída: a saída a seguir é típica para registrar uma EC2 instância.

```
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.
```

Mais informações

Para obter mais informações, consulte [Registrar uma instância com uma AWS OpsWorks pilha no Guia do AWS OpsWorks usuário](#).

- Para API obter detalhes, consulte [Registrar](#) na Referência de AWS CLI Comandos.

set-load-based-auto-scaling

O código de exemplo a seguir mostra como usar `set-load-based-auto-scaling`.

AWS CLI

Para definir a configuração de escalabilidade baseada em carga para uma camada

O exemplo a seguir ativa o dimensionamento baseado em carga para uma camada especificada e define a configuração dessa camada. Você deve usar `create-instance` para adicionar instâncias baseadas em carga à camada.

```
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
```

O exemplo coloca as configurações de limite de aumento de escala em um arquivo separado no diretório de trabalho chamado `upscale.json`, que contém o seguinte.

```
{  
  "InstanceCount": 2,  
  "ThresholdsWaitTime": 3,  
  "IgnoreMetricsTime": 3,  
  "CpuThreshold": 85,  
  "MemoryThreshold": 85,  
  "LoadThreshold": 85  
}
```

O exemplo coloca as configurações de limite de redução de escala em um arquivo separado no diretório de trabalho chamado `downscale.json`, que contém o seguinte.

```
{  
  "InstanceCount": 2,  
  "ThresholdsWaitTime": 3,  
  "IgnoreMetricsTime": 3,  
  "CpuThreshold": 35,  
  "MemoryThreshold": 30,  
  "LoadThreshold": 30  
}
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Usando o escalonamento automático baseado em carga no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [SetLoadBasedAutoScaling](#) na Referência de AWS CLI Comandos.

set-permission

O código de exemplo a seguir mostra como usar `set-permission`.

AWS CLI

Para conceder níveis de permissão por pilha AWS OpsWorks

Quando você importa um usuário do AWS Identity and Access Management (IAM) AWS OpsWorks por meio de uma chamada `create-user-profile`, o usuário tem somente as permissões concedidas pelas IAM políticas anexadas. Você pode conceder AWS OpsWorks permissões modificando as políticas de um usuário. No entanto, geralmente é mais fácil importar um usuário e depois usar o `set-permission` comando para conceder ao usuário um dos níveis de permissão padrão para cada pilha à qual o usuário precisará acessar.

O exemplo a seguir concede permissão para a pilha especificada para um usuário, que é identificado pelo Amazon Resource Name (ARN). O exemplo concede ao usuário um nível de permissões de gerenciamento, com `sudo` e SSH privilégios nas instâncias da pilha.

```
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
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Conceder permissões por pilha AWS OpsWorks aos usuários no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [SetPermission](#) na Referência de AWS CLI Comandos.

set-time-based-auto-scaling

O código de exemplo a seguir mostra como usar `set-time-based-auto-scaling`.

AWS CLI

Para definir a configuração de escalabilidade baseada em tempo para uma camada

O exemplo a seguir define a configuração baseada em tempo para uma instância especificada. Você deve primeiro usar `create-instance` para adicionar a instância à camada.

```
aws opsworks --region us-east-1 set-time-based-auto-scaling --instance-id 69b6237c-08c0-4edb-a6af-78f3d01cedf2 --auto-scaling-schedule file://schedule.json
```

O exemplo coloca a agenda em um arquivo separado no diretório de trabalho chamados `schedule.json`. Neste exemplo, a instância fica ativada por algumas horas por volta do meio-dia UTC (Horário Universal Coordenado) na segunda e na terça.

```
{
  "Monday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  },
  "Tuesday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  }
}
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Usando o escalonamento automático baseado em tempo](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [SetTimeBasedAutoScaling](#) na Referência de AWS CLI Comandos.

start-instance

O código de exemplo a seguir mostra como usar `start-instance`.

AWS CLI

Para iniciar uma instância

O `start-instance` comando a seguir inicia uma instância específica 24 horas por dia, 7 dias por semana.

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

Saída: Nenhuma. Use `describe-instances` para verificar o status da instância.

Dica Você pode iniciar cada instância offline em uma pilha com um comando chamando `start-stack`.

Mais informações

Para obter mais informações, consulte Iniciando, interrompendo e reiniciando manualmente instâncias 24 horas por dia, 7 dias por semana, no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [StartInstance](#) na Referência de AWS CLI Comandos.

start-stack

O código de exemplo a seguir mostra como usar `start-stack`.

AWS CLI

Para iniciar as instâncias de uma pilha

O exemplo a seguir inicia todas as instâncias 24 horas por dia, 7 dias por semana de uma pilha. Para iniciar uma instância específica, use `start-instance`.

```
aws opsworks --region us-east-1 start-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Como iniciar uma instância no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [StartStack](#) na Referência de AWS CLI Comandos.

stop-instance

O código de exemplo a seguir mostra como usar `stop-instance`.

AWS CLI

Para parar uma instância

O exemplo a seguir interrompe uma instância especificada, que é identificada por seu ID de instância. Você pode obter um ID de instância acessando a página de detalhes da instância no AWS OpsWorks console ou executando o `describe-instances` comando.

```
aws opsworks stop-instance --region us-east-1 --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

Você pode reiniciar uma instância interrompida chamando `start-instance` ou excluindo a instância por meio de uma chamada `delete-instance`.

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Como interromper uma instância](#) no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [StopInstance](#) na Referência de AWS CLI Comandos.

stop-stack

O código de exemplo a seguir mostra como usar `stop-stack`.

AWS CLI

Para interromper as instâncias de uma pilha

O exemplo a seguir interrompe todas as instâncias 24 horas por dia, 7 dias por semana de uma pilha. Para interromper uma instância específica, use `stop-instance`.

```
aws opsworks --region us-east-1 stop-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Saída: Sem saída.

Mais informações

Para obter mais informações, consulte [Como interromper uma instância](#) no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [StopStack](#) na Referência de AWS CLI Comandos.

unassign-instance

O código de exemplo a seguir mostra como usar `unassign-instance`.

AWS CLI

Para cancelar a atribuição de uma instância registrada de suas camadas

O `unassign-instance` comando a seguir cancela a atribuição de uma instância de suas camadas anexadas.

```
aws opsworks --region us-east-1 unassign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Cancelar a atribuição de uma instância registrada](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [UnassignInstance](#) na Referência de AWS CLI Comandos.

unassign-volume

O código de exemplo a seguir mostra como usar `unassign-volume`.

AWS CLI

Para cancelar a atribuição de um volume de sua instância

O exemplo a seguir cancela a atribuição de um volume registrado do Amazon Elastic Block Store EBS (Amazon) de sua instância. O volume é identificado por seu ID de volume, que é atribuído quando você registra o GUID volume em uma pilha, não pelo ID de volume do Amazon Elastic Compute Cloud (AmazonEC2). AWS OpsWorks

```
aws opsworks --region us-east-1 unassign-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Cancelar a atribuição de EBS volumes da Amazon](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [UnassignVolume](#) na Referência de AWS CLI Comandos.

update-app

O código de exemplo a seguir mostra como usar `update-app`.

AWS CLI

Para atualizar um aplicativo

O exemplo a seguir atualiza um aplicativo especificado para alterar seu nome.

```
aws opsworks --region us-east-1 update-app --app-id 26a61ead-d201-47e3-b55c-2a7c666942f8 --name NewAppName
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Editando aplicativos no Guia AWS OpsWorks do usuário](#).

- Para API obter detalhes, consulte [UpdateApp](#) na Referência de AWS CLI Comandos.

update-elastic-ip

O código de exemplo a seguir mostra como usar `update-elastic-ip`.

AWS CLI

Para atualizar um nome de endereço IP elástico

O exemplo a seguir atualiza o nome de um endereço IP elástico especificado.

```
aws opsworks --region us-east-1 update-elastic-ip --elastic-ip 54.148.130.96 --name NewIPName
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Gerenciamento de recursos no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [UpdateElasticIp](#) na Referência de AWS CLI Comandos.

update-instance

O código de exemplo a seguir mostra como usar `update-instance`.

AWS CLI

Para atualizar uma instância

O exemplo a seguir atualiza o tipo de uma instância especificada.

```
aws opsworks --region us-east-1 update-instance --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f --instance-type c3.xlarge
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Edição da configuração da instância no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [UpdateInstance](#) na Referência de AWS CLI Comandos.

update-layer

O código de exemplo a seguir mostra como usar `update-layer`.

AWS CLI

Para atualizar uma camada

O exemplo a seguir atualiza uma camada especificada para usar instâncias EBS otimizadas pela Amazon.

```
aws opsworks --region us-east-1 update-layer --layer-  
id 888c5645-09a5-4d0e-95a8-812ef1db76a4 --use-eks-optimized-instances
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Editando a configuração de uma OpsWorks camada no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [UpdateLayer](#) na Referência de AWS CLI Comandos.

update-my-user-profile

O código de exemplo a seguir mostra como usar `update-my-user-profile`.

AWS CLI

Para atualizar o perfil de um usuário

O exemplo a seguir atualiza o perfil do `development` usuário para usar uma chave SSH pública especificada. As AWS credenciais do usuário são representadas pelo `development` perfil no `credentials` arquivo (`~\.aws\credentials`) e a chave está em um `.pem` arquivo no diretório de trabalho.

```
aws opsworks --region us-east-1 --profile development update-my-user-profile --ssh-  
public-key file://development_key.pem
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Edição das configurações AWS OpsWorks do usuário no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [UpdateMyUserProfile](#) na Referência de AWS CLI Comandos.

update-rds-db-instance

O código de exemplo a seguir mostra como usar `update-rds-db-instance`.

AWS CLI

Para atualizar uma instância de RDS banco de dados Amazon registrada

O exemplo a seguir atualiza o valor da senha mestra de uma RDS instância da Amazon. Observe que esse comando não altera a senha mestra da RDS instância, apenas a senha que você fornece AWS OpsWorks. Se essa senha não corresponder à senha da RDS instância, seu aplicativo não conseguirá se conectar ao banco de dados.

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte Registro de RDS instâncias da Amazon com uma pilha no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [UpdateRdsDbInstance](#) na Referência de AWS CLI Comandos.

update-volume

O código de exemplo a seguir mostra como usar `update-volume`.

AWS CLI

Para atualizar um volume registrado

O exemplo a seguir atualiza o ponto de montagem de um volume registrado do Amazon Elastic Block Store (AmazonEBS). O volume é identificado por seu ID de volume, que é atribuído ao GUID volume quando você o registra em uma pilha, não pelo ID de volume do Amazon Elastic Compute Cloud (AmazonEC2). AWS OpsWorks :

```
aws opsworks --region us-east-1 update-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df --mount-point /mnt/myvol
```

Saída: Nenhuma.

Mais informações

Para obter mais informações, consulte [Atribuição de EBS volumes da Amazon](#) a uma instância no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [UpdateVolume](#) na Referência de AWS CLI Comandos.

AWS OpsWorks CM exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS OpsWorks CM.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-node

O código de exemplo a seguir mostra como usar `associate-node`.

AWS CLI

Para associar nós

O `associate-node` comando a seguir associa um nó nomeado `i-44de882p` a um servidor Chef Automate chamado `automate-06`, o que significa que o `automate-06` servidor gerencia o nó e comunica comandos de receita ao nó por meio do software `chef-client` agente que é instalado no nó pelo comando `associate-node`. Os nomes de nós válidos são EC2 instânciasIDs. :

```
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'"
```

A saída retornada pelo comando é semelhante à seguinte. Saída:

```
{
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"
}
```

Mais informações

Para obter mais informações, consulte [Adicionando nós automaticamente ao Chef Automate no Guia do AWS OpsWorks usuário](#). AWS OpsWorks

- Para API obter detalhes, consulte [AssociateNode](#) na Referência de AWS CLI Comandos.

create-backup

O código de exemplo a seguir mostra como usar `create-backup`.

AWS CLI

Para criar backups

O `create-backup` comando a seguir inicia um backup manual de um servidor Chef Automate nomeado `automate-06` na `us-east-1` região. O comando adiciona uma mensagem descritiva ao backup no `--description` parâmetro.

```
aws opsworks-cm create-backup \
  --server-name 'automate-06' \
  --description "state of my infrastructure at launch"
```

A saída mostra informações semelhantes às seguintes sobre o novo backup.

Saída:

```
{
  "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"
  }
],
}

```

Para obter mais informações, consulte [Fazer backup e restaurar um AWS OpsWorks para o Chef Automate Server](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [CreateBackup](#) na Referência de AWS CLI Comandos.

create-server

O código de exemplo a seguir mostra como usar `create-server`.

AWS CLI

Para criar um servidor

O `create-server` exemplo a seguir cria um novo servidor Chef Automate chamado `automate-06` em sua região padrão. Observe que os padrões são usados para a maioria das outras configurações, como o número de backups a serem retidos e os horários de início da manutenção e do backup. Antes de executar um `create-server` comando, preencha os pré-requisitos em [Introdução ao Chef Automate no Guia do usuário do AWS Opsworks 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"
```

A saída mostra informações semelhantes às seguintes sobre o novo servidor:

```
{
  "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" ]  
  }  
}
```

Para obter mais informações, consulte [UpdateServer](#) API Referência do AWS OpsWorks Chef Automate.

- Para API obter detalhes, consulte [CreateServer](#) na Referência de AWS CLI Comandos.

delete-backup

O código de exemplo a seguir mostra como usar `delete-backup`.

AWS CLI

Para excluir backups

O `delete-backup` comando a seguir exclui um backup manual ou automatizado de um servidor Chef Automate, identificado pelo ID de backup. Esse comando é útil quando você está se aproximando do número máximo de backups que você pode salvar ou deseja minimizar os custos de armazenamento do Amazon S3. :

```
aws opsworks-cm delete-backup --backup-id "automate-06-2016-11-19T23:42:40.240Z"
```

A saída mostra se a exclusão do backup foi bem-sucedida.

Mais informações

Para obter mais informações, consulte [Fazer backup e restaurar um AWS OpsWorks para o Chef Automate Server](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DeleteBackup](#) na Referência de AWS CLI Comandos.

delete-server

O código de exemplo a seguir mostra como usar `delete-server`.

AWS CLI

Para excluir servidores

O `delete-server` comando a seguir exclui um servidor Chef Automate, identificado pelo nome do servidor. Depois que o servidor é excluído, ele não é mais retornado pelas `DescribeServer` solicitações. :

```
aws opsworks-cm delete-server --server-name "automate-06"
```

A saída mostra se a exclusão do servidor foi bem-sucedida.

Mais informações

Para obter mais informações, consulte [Excluir um AWS OpsWorks para o Chef Automate Server](#) no Guia do AWS OpsWorks Usuário.

- Para API obter detalhes, consulte [DeleteServer](#) na Referência de AWS CLI Comandos.

describe-account-attributes

O código de exemplo a seguir mostra como usar `describe-account-attributes`.

AWS CLI

Como descrever os atributos da conta

O `describe-account-attributes` comando a seguir retorna informações sobre o uso de recursos do Chef Automate AWS OpsWorks pela sua conta. :

```
aws opsworks-cm describe-account-attributes
```

A saída para cada entrada de atributo de conta retornada pelo comando é semelhante à seguinte. Saída:

```
{
  "Attributes": [
    {
      "Maximum": 5,
      "Name": "ServerLimit",
      "Used": 2
    }
  ]
}
```

Mais informações

Para obter mais informações, consulte [DescribeAccountAttributes](#) a APIReferência do AWS OpsWorks Chef Automate.

- Para API obter detalhes, consulte [DescribeAccountAttributes](#) na Referência de AWS CLI Comandos.

describe-backups

O código de exemplo a seguir mostra como usar `describe-backups`.

AWS CLI

Para descrever os backups

O `describe-backups` comando a seguir retorna informações sobre todos os backups associados à sua conta na sua região padrão.

```
aws opsworks-cm describe-backups
```

A saída de cada entrada de backup retornada pelo comando é semelhante à seguinte.

Saída:

```
{
  "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"
    }
],
}

```

Para obter mais informações, consulte [Fazer backup e restaurar um AWS OpsWorks para o Chef Automate Server](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [DescribeBackups](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para descrever eventos

O `describe-events` exemplo a seguir retorna informações sobre todos os eventos associados ao servidor Chef Automate especificado.

```

aws opsworks-cm describe-events \
  --server-name 'automate-06'

```

A saída de cada entrada de evento retornada pelo comando é semelhante ao exemplo a seguir:

```

{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Dicas gerais de solução de problemas](#) no Guia AWS OpsWorks do usuário.

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-node-association-status

O código de exemplo a seguir mostra como usar `describe-node-association-status`.

AWS CLI

Para descrever o status da associação de nós

O `describe-node-association-status` comando a seguir retorna o status de uma solicitação para associar um nó a um servidor Chef Automate chamado `automate-06`.

```
aws opsworks-cm describe-node-association-status --server-
name "automate-06" --node-association-status-token "AfLJKL+/
GoKLZJBdDQEx0065CDi57b1Qe9nKM8joSok0pQ9xr8DqApBN9/106sLdSvLfDEKkEx+eoCHvjowHa0s="
```

A saída para cada entrada de atributo de conta retornada pelo comando é semelhante à seguinte. Saída:

```
{
  "NodeAssociationStatus": "IN_PROGRESS"
}
```

Mais informações

Para obter mais informações, consulte `DescribeNodeAssociationStatus` a API Referência do AWS OpsWorks Chef Automate.

- Para API obter detalhes, consulte [DescribeNodeAssociationStatus](#) na Referência de AWS CLI Comandos.

describe-servers

O código de exemplo a seguir mostra como usar `describe-servers`.

AWS CLI

Para descrever servidores

O `describe-servers` comando a seguir retorna informações sobre todos os servidores associados à sua conta e em sua região padrão. :

```
aws opsworks-cm describe-servers
```

A saída para cada entrada do servidor retornada pelo comando é semelhante à seguinte. Saída:

```
{
  "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" ]
  }
]
}

```

Mais informações

Para obter mais informações, consulte [DescribeServers](#) o AWS OpsWorks API Guia do Chef Automate.

- Para API obter detalhes, consulte [DescribeServers](#) na Referência de AWS CLI Comandos.

disassociate-node

O código de exemplo a seguir mostra como usar `disassociate-node`.

AWS CLI

Para desassociar nós

O `disassociate-node` comando a seguir desassocia um nó chamado `i-44de882p`, removendo o nó do gerenciamento por um servidor Chef Automate chamado `automate-06`. Os nomes de nós válidos são IDs de instâncias EC2.

```

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'"

```

A saída retornada pelo comando é semelhante à seguinte. Saída:

```

{
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"
}

```

Mais informações

Para obter mais informações, consulte [Excluir um AWS OpsWorks para o Chef Automate Server](#) no Guia do AWS OpsWorks Usuário.

- Para API obter detalhes, consulte [DisassociateNode](#) na Referência de AWS CLI Comandos.

restore-server

O código de exemplo a seguir mostra como usar `restore-server`.

AWS CLI

Para restaurar um servidor

O `restore-server` comando a seguir executa uma restauração local de um servidor Chef Automate nomeado `automate-06` em sua região padrão a partir de um backup com um ID de `automate-06-2016-11-22T16:13:27.998Z`. A restauração de um servidor restaura as conexões com os nós que o servidor Chef Automate estava gerenciando no momento em que o backup especificado foi executado.

```
aws opsworks-cm restore-server --backup-id "Automate-06-2016-11-22T 16:13:27.998 Z" --nome do servidor "automate-06"
```

A saída é somente o ID do comando. Saída:

```
(None)
```

Mais informações

Para obter mais informações, consulte [Restaurar uma falha AWS OpsWorks para o Chef Automate Server](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [RestoreServer](#) na Referência de AWS CLI Comandos.

start-maintenance

O código de exemplo a seguir mostra como usar `start-maintenance`.

AWS CLI

Para iniciar a manutenção

O `start-maintenance` exemplo a seguir inicia manualmente a manutenção no servidor Chef Automate ou Puppet Enterprise especificado em sua região padrão. Esse comando é útil se uma tentativa anterior de manutenção automatizada falhar e a causa subjacente da falha de manutenção tiver sido resolvida.

```
aws opsworks-cm start-maintenance \  
  --server-name 'automate-06'
```

Saída:

```
{  
  "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-EXAMPLEvr8gjfk5f.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"  
  }  
}
```

Para obter mais informações, consulte [Manutenção do sistema \(servidores Puppet Enterprise\)](#) ou [Manutenção do sistema \(servidores Chef Automate\)](#) no Guia do AWS OpsWorks usuário.

- Para API obter detalhes, consulte [StartMaintenance](#) na Referência de AWS CLI Comandos.

update-server-engine-attributes

O código de exemplo a seguir mostra como usar `update-server-engine-attributes`.

AWS CLI

Para atualizar os atributos do mecanismo do servidor

O `update-server-engine-attributes` comando a seguir atualiza o valor do atributo `CHEF_PIVOTAL_KEY` do mecanismo para um servidor Chef Automate chamado `automate-06`. Atualmente, não é possível alterar o valor de outros atributos do mecanismo.

```
aws opsworks-cm update-server-engine-attributes \  
  --attribute-name CHEF_PIVOTAL_KEY \  
  --attribute-value "new key value" \  
  --server-name "automate-06"
```

A saída mostra informações semelhantes às seguintes sobre o servidor atualizado.

```
{  
  "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" ]
  }
}

```

Para obter mais informações, consulte [UpdateServerEngineAttributes](#) a API Referência do AWS OpsWorks Chef Automate.

- Para API obter detalhes, consulte [UpdateServerEngineAttributes](#) na Referência de AWS CLI Comandos.

update-server

O código de exemplo a seguir mostra como usar `update-server`.

AWS CLI

Para atualizar um servidor

O `update-server` comando a seguir atualiza a hora de início da manutenção do servidor Chef Automate especificado em sua região padrão. O `--preferred-maintenance-window` parâmetro é adicionado para alterar o dia e a hora de início da manutenção do servidor para segundas-feiras às 9h15. UTC. :

```

aws opsworks-cm update-server \
  --server-name "automate-06" \
  --preferred-maintenance-window "Mon:09:15"

```

A saída mostra informações semelhantes às seguintes sobre o servidor atualizado.

```

{
  "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" ]
  }
}

```

Para obter mais informações, consulte [UpdateServer](#) a API Referência do AWS OpsWorks Chef Automate.

- Para API obter detalhes, consulte [UpdateServer](#) na Referência de AWS CLI Comandos.

Exemplos de organizações usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Organizations.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-handshake

O código de exemplo a seguir mostra como usar `accept-handshake`.

AWS CLI

Para aceitar um aperto de mão de outra conta

Bill, proprietário de uma organização, já havia convidado a conta de Juan para se juntar à sua organização. O exemplo a seguir mostra a conta de Juan aceitando o aperto de mão e, assim, concordando com o convite.

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

A saída mostra o seguinte:

```
{
  "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"
  }
}

```

- Para API obter detalhes, consulte [AcceptHandshake](#) na Referência de AWS CLI Comandos.

attach-policy

O código de exemplo a seguir mostra como usar `attach-policy`.

AWS CLI

Como anexar uma política a uma raiz, unidade operacional ou conta

Exemplo 1

O exemplo a seguir mostra como anexar uma política de controle de serviço (SCP) a uma OU:

```
aws organizations attach-policy
```



```
--policy-id p-examplepolicyid111  
--target-id ou-examplerootid111-exampleoid111
```

Exemplo 2

O seguinte exemplo mostra como anexar uma política de controle de serviços a uma conta:

```
aws organizations attach-policy  
--policy-id p-examplepolicyid111  
--target-id 333333333333
```

- Para API obter detalhes, consulte [AttachPolicy](#) na Referência de AWS CLI Comandos.

cancel-handshake

O código de exemplo a seguir mostra como usar `cancel-handshake`.

AWS CLI

Para cancelar um aperto de mão enviado de outra conta

Bill já havia enviado um convite para a conta de Susan para se juntar à sua organização. Ele muda de ideia e decide cancelar o convite antes que Susan o aceite. O exemplo a seguir mostra o cancelamento de Bill:

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

A saída inclui um objeto de aperto de mão que mostra que o estado agora CANCELED é:

```
{  
  "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
}
}

```

- Para API obter detalhes, consulte [CancelHandshake](#) na Referência de AWS CLI Comandos.

create-account

O código de exemplo a seguir mostra como usar create-account.

AWS CLI

Como criar uma conta de membro que automaticamente faça parte da organização

O exemplo a seguir mostra como criar uma conta de membro em uma organização. A conta de membro é configurada com o nome Production Account e o endereço de e-mail susan@example.com. Organizations cria automaticamente uma IAM função usando o nome padrão de OrganizationAccountAccessRole porque o roleName parâmetro não está especificado. Além disso, a configuração que permite que IAM usuários ou funções com permissões suficientes acessem os dados de faturamento da conta é definida como o valor padrão de ALLOW porque o lamUserAccessToBilling parâmetro não foi especificado. Organizations envia automaticamente a Susan um e-mail de “Bem-vindo a AWS”:

```
aws organizations create-account --email susan@example.com --account-name "Production Account"
```

A saída inclui um objeto de solicitação que mostra que o status agora é IN_PROGRESS:

```
{
  "CreateAccountStatus": {
    "State": "IN_PROGRESS",
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

Posteriormente, você pode consultar o status atual da solicitação fornecendo o valor de resposta Id ao describe-create-account-status comando como o valor do create-account-request-id parâmetro.

Para obter mais informações, consulte Criando uma AWS conta em sua organização no Guia do Usuário do AWS Organizations.

- Para API obter detalhes, consulte [CreateAccount](#) na Referência de AWS CLI Comandos.

create-organization

O código de exemplo a seguir mostra como usar create-organization.

AWS CLI

Exemplo 1: como criar uma organização

Bill quer criar uma organização usando as credenciais da conta 111111111111. O exemplo a seguir mostra que a conta se torna a conta principal na nova organização. Como ele não especificou um conjunto de recursos, a nova organização usa como padrão todos os recursos habilitados e as políticas de controle de serviços são habilitadas na raiz.

```
aws organizations create-organization
```

A saída inclui um objeto de organização com detalhes sobre a nova organização:

```
{
  "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"
  }
}
```

Exemplo 2: como criar uma organização apenas com os recursos de faturamento consolidados

O seguinte exemplo cria uma organização compatível apenas com os recursos de faturamento consolidados:

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

A saída inclui um objeto de organização com detalhes sobre a nova organização:

```
{
  "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"
    }
}

```

Para obter informações, consulte [Criar uma organização](#) no Guia do usuário do AWS Organizations.

- Para API obter detalhes, consulte [CreateOrganization](#) na Referência de AWS CLI Comandos.

create-organizational-unit

O código de exemplo a seguir mostra como usar `create-organizational-unit`.

AWS CLI

Como criar uma unidade organizacional em uma unidade organizacional raiz ou pai

O seguinte exemplo mostra como criar uma UO chamada AccountingOU:

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

A saída inclui um `organizationalUnit` objeto com detalhes sobre a nova OU:

```

{
  "OrganizationalUnit": {
    "Id": "ou-examplerootid111-exampleoid111",
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleoid111",
    "Name": "AccountingOU"
  }
}

```

- Para API obter detalhes, consulte [CreateOrganizationalUnit](#) na Referência de AWS CLI Comandos.

create-policy

O código de exemplo a seguir mostra como usar `create-policy`.

AWS CLI

Exemplo 1: Para criar uma política com um arquivo de origem de texto para a JSON política

O exemplo a seguir mostra como criar uma política de controle de serviço (SCP) chamada `AllowAllS3Actions`. O conteúdo da política provém de um arquivo chamado `policy.json` presente no computador local.

```
aws organizations create-policy --content file://policy.json --  
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows  
delegation of all S3 actions"
```

A saída inclui um objeto de política com detalhes sobre a nova política:

```
{  
  "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"  
    }  
  }  
}
```

Exemplo 2: Para criar uma política com uma JSON política como parâmetro

O exemplo a seguir mostra como criar o mesmo SCP, desta vez incorporando o conteúdo da política como uma JSON string no parâmetro. A string deve ser recuada com barras invertidas antes das aspas duplas para garantir que ela seja tratada como literal no parâmetro (que está entre aspas duplas):

```
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"
```

Para obter mais informações sobre como criar e usar políticas em sua organização, consulte Gerenciamento de políticas organizacionais no Guia do usuário do AWS Organizations.

- Para API obter detalhes, consulte [CreatePolicy](#) na Referência de AWS CLI Comandos.

decline-handshake

O código de exemplo a seguir mostra como usar `decline-handshake`.

AWS CLI

Para recusar um aperto de mão enviado de outra conta

O exemplo a seguir mostra que Susan, uma administradora que é proprietária da conta 2222222222, recusa um convite para se juntar à organização de Bill. A `DeclineHandshake` operação retorna um objeto de aperto de mão, mostrando que o estado agora `DECLINED` é:

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

A saída inclui um objeto de aperto de mão que mostra o novo estado de `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"
}
}

```

- Para API obter detalhes, consulte [DeclineHandshake](#) na Referência de AWS CLI Comandos.

delete-organization

O código de exemplo a seguir mostra como usar delete-organization.

AWS CLI

Como excluir uma organização

O exemplo a seguir mostra como excluir uma organização. Você deve ser administrador da conta principal na organização para poder realizar essa operação. O exemplo pressupõe que você removeu anteriormente todas as contas e políticas dos membros da organização: OUs


```
aws organizations delete-organization
```

- Para API obter detalhes, consulte [DeleteOrganization](#) na Referência de AWS CLI Comandos.

delete-organizational-unit

O código de exemplo a seguir mostra como usar delete-organizational-unit.

AWS CLI

Como excluir uma UO

O exemplo a seguir mostra como excluir uma UO. O exemplo pressupõe que você removeu anteriormente todas as contas e outras OUs da OU:

```
aws organizations delete-organizational-unit --organizational-unit-id ou-examplerootid111-exampleoid111
```

- Para API obter detalhes, consulte [DeleteOrganizationalUnit](#) na Referência de AWS CLI Comandos.

delete-policy

O código de exemplo a seguir mostra como usar delete-policy.

AWS CLI

Como excluir uma política

O exemplo a seguir mostra como excluir uma política de uma organização. O exemplo pressupõe que você já separou a política de todas as entidades:

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

describe-account

O código de exemplo a seguir mostra como usar describe-account.

AWS CLI

Para obter os detalhes sobre uma conta

O exemplo a seguir mostra como solicitar detalhes sobre uma conta:

```
aws organizations describe-account --account-id 555555555555
```

A saída mostra um objeto de conta com os detalhes sobre a conta:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [DescribeAccount](#) na Referência de AWS CLI Comandos.

describe-create-account-status

O código de exemplo a seguir mostra como usar `describe-create-account-status`.

AWS CLI

Para obter o status mais recente sobre uma solicitação para criar uma conta

O exemplo a seguir mostra como solicitar o status mais recente de uma solicitação anterior para criar uma conta em uma organização. O `--request-id` especificado vem da resposta da chamada original para `create-account`. A solicitação de criação de conta mostra pelo campo de status que Organizations concluiu com êxito a criação da conta.

Comando:

```
aws organizations describe-create-account-status --create-account-request-id car-
examplecreateaccountrequestid111
```

Saída:

```
{
  "CreateAccountStatus": {
    "State": "SUCCEEDED",
    "AccountId": "555555555555",
    "AccountName": "Beta account",
    "RequestedTimestamp": 1470684478.687,
    "CompletedTimestamp": 1470684532.472,
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

- Para API obter detalhes, consulte [DescribeCreateAccountStatus](#) na Referência de AWS CLI Comandos.

describe-handshake

O código de exemplo a seguir mostra como usar `describe-handshake`.

AWS CLI

Para obter informações sobre um aperto de mão

O exemplo a seguir mostra como solicitar detalhes sobre um aperto de mão. O ID do handshake vem da chamada original para `InviteAccountToOrganization`, ou de uma chamada para `ListHandshakesForAccount` ou `ListHandshakesForOrganization`:

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

A saída inclui um objeto de aperto de mão que tem todos os detalhes sobre o aperto de mão solicitado:

```
{
  "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"
}
}

```

- Para API obter detalhes, consulte [DescribeHandshake](#) na Referência de AWS CLI Comandos.

describe-organization

O código de exemplo a seguir mostra como usar describe-organization.

AWS CLI

Para obter informações sobre a organização atual

O exemplo a seguir mostra como solicitar detalhes sobre uma organização:

```
aws organizations describe-organization
```

A saída inclui um objeto de organização que tem os detalhes sobre a organização:

```
{
  "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"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [DescribeOrganization](#) na Referência de AWS CLI Comandos.

describe-organizational-unit

O código de exemplo a seguir mostra como usar describe-organizational-unit.

AWS CLI

Para obter informações sobre uma OU

O describe-organizational-unit exemplo a seguir solicita detalhes sobre uma OU.

```
aws organizations describe-organizational-unit \
  --organizational-unit-id ou-exampleorgid111-exampleoid111
```

Saída:

```
{
  "OrganizationalUnit": {
    "Name": "Accounting Group",
    "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-
    exempleroottid111-exampleoid111",
    "Id": "ou-exempleroottid111-exampleoid111"
  }
}
```

- Para API obter detalhes, consulte [DescribeOrganizationalUnit](#) na Referência de AWS CLI Comandos.

describe-policy

O código de exemplo a seguir mostra como usar `describe-policy`.

AWS CLI

Como obter informações sobre uma política

O seguinte exemplo mostra como solicitar informações sobre uma política:

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

A saída inclui um objeto de política que contém detalhes sobre a política:

```
{
  "Policy": {
    "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n
    {\n    \"Effect\": \"Allow\",\n    \"Action\": \"*\",\n    \"Resource\":
    \"*\"\n  }]\n}",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::111111111111:policy/o-
      exampleorgid/service_control_policy/p-examplepolicyid111",
      "Type": "SERVICE_CONTROL_POLICY",
      "Id": "p-examplepolicyid111",
      "AwsManaged": false,
      "Name": "AllowAllS3Actions",
      "Description": "Enables admins to delegate S3 permissions"
    }
  }
}
```

```
}
```

- Para API obter detalhes, consulte [DescribePolicy](#) na Referência de AWS CLI Comandos.

detach-policy

O código de exemplo a seguir mostra como usar `detach-policy`.

AWS CLI

Como separar uma política de uma raiz, UO ou conta

O seguinte exemplo mostra como separar uma política de uma UO:

```
aws organizations detach-policy --target-id ou-examplerootid111-exampleouid111 --  
policy-id p-examplepolicyid111
```

- Para API obter detalhes, consulte [DetachPolicy](#) na Referência de AWS CLI Comandos.

disable-policy-type

O código de exemplo a seguir mostra como usar `disable-policy-type`.

AWS CLI

Para desativar um tipo de política em uma raiz

O exemplo a seguir mostra como desabilitar o tipo de política de controle de serviço (SCP) em uma raiz:

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-  
type SERVICE_CONTROL_POLICY
```

A saída mostra que o elemento de `PolicyTypes` resposta não inclui mais `SERVICE _ CONTROL _ POLICY`:

```
{  
  "Root": {  
    "PolicyTypes": [],  
    "Name": "Root",  
    "Id": "r-examplerootid111",
```

```

        "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
    exemplerooid111"
    }
}

```

- Para API obter detalhes, consulte [DisablePolicyType](#) na Referência de AWS CLI Comandos.

enable-all-features

O código de exemplo a seguir mostra como usar `enable-all-features`.

AWS CLI

Para habilitar todos os recursos em uma organização

Este exemplo mostra o administrador solicitando que todas as contas convidadas da organização aprove todos os recursos habilitados na organização. AWS Organizations envia um e-mail para o endereço registrado em cada conta de membro convidado solicitando que o proprietário aprove a alteração em todos os recursos aceitando o aperto de mão enviado. Depois que todas as contas de membros convidados aceitarem o handshake, o administrador da organização poderá finalizar a alteração em todos os recursos, e aqueles com as permissões apropriadas poderão criar políticas e aplicá-las às raízes e às contas: OUs

```
aws organizations enable-all-features
```

O resultado é um objeto de aperto de mão que é enviado para aprovação de todas as contas de membros convidados:

```

{
    "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"
  }
}
```

- Para API obter detalhes, consulte [EnableAllFeatures](#) na Referência de AWS CLI Comandos.

enable-policy-type

O código de exemplo a seguir mostra como usar `enable-policy-type`.

AWS CLI

Para habilitar o uso de um tipo de política em uma raiz

O exemplo a seguir mostra como habilitar o tipo de política de controle de serviço (SCP) em uma raiz:

```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-type SERVICE_CONTROL_POLICY
```

A saída mostra um objeto raiz com um elemento de `policyTypes` resposta mostrando que agora SCPs estão habilitados:

```
{
  "Root": {
    "PolicyTypes": [
      {
        "Status":"ENABLED",
        "Type":"SERVICE_CONTROL_POLICY"
      }
    ],
    "Id": "r-examplerootid111",
    "Name": "Root",
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-examplerootid111"
  }
}
```

```
}
}
```

- Para API obter detalhes, consulte [EnablePolicyType](#) na Referência de AWS CLI Comandos.

invite-account-to-organization

O código de exemplo a seguir mostra como usar `invite-account-to-organization`.

AWS CLI

Para convidar uma conta para participar de uma organização

O exemplo a seguir mostra a conta principal de propriedade de `bill@example.com` convidando a conta de propriedade de `juan@example.com` para se juntar a uma organização:

```
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."
```

O resultado inclui uma estrutura de handshake que mostra o que é enviado para a conta convidada:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [InviteAccountToOrganization](#) na Referência de AWS CLI Comandos.

leave-organization

O código de exemplo a seguir mostra como usar `leave-organization`.

AWS CLI

Para sair de uma organização como conta de membro

O exemplo a seguir mostra o administrador de uma conta de membro solicitando sair da organização da qual é membro atualmente:

```
aws organizations leave-organization
```

- Para API obter detalhes, consulte [LeaveOrganization](#) na Referência de AWS CLI Comandos.

list-accounts-for-parent

O código de exemplo a seguir mostra como usar `list-accounts-for-parent`.

AWS CLI

Para recuperar uma lista de todas as contas em uma raiz principal ou OU especificada

O exemplo a seguir mostra como solicitar uma lista das contas em uma OU:

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-exampleouid111
```

A saída inclui uma lista de objetos de resumo da conta.

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListAccountsForParent](#) na Referência de AWS CLI Comandos.

list-accounts

O código de exemplo a seguir mostra como usar `list-accounts`.

AWS CLI

Como recuperar uma lista de todas as contas de uma organização

O seguinte exemplo mostra como solicitar uma lista das contas de uma organização:

```
aws organizations list-accounts
```

A saída inclui uma lista de objetos de resumo da conta.

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListAccounts](#) na Referência de AWS CLI Comandos.

list-children

O código de exemplo a seguir mostra como usar `list-children`.

AWS CLI

Para recuperar as contas secundárias e OUs de uma OU principal ou raiz

O exemplo a seguir mostra como listar a raiz ou OU que contém essa conta 444444444444:

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-
examplerootid111-exampleoid111
```

A saída mostra os dois filhos OUs contidos pelo pai:

```

{
    "Children": [
        {
            "Id": "ou-examplerootid111-exampleoid111",
            "Type": "ORGANIZATIONAL_UNIT"
        },
        {

```

```

        "Id": "ou-examplerootid111-exampleouid222",
        "Type": "ORGANIZATIONAL_UNIT"
    }
]
}

```

- Para API obter detalhes, consulte [ListChildren](#) na Referência de AWS CLI Comandos.

list-create-account-status

O código de exemplo a seguir mostra como usar `list-create-account-status`.

AWS CLI

Exemplo 1: Para recuperar uma lista das solicitações de criação de conta feitas na organização atual

O exemplo a seguir mostra como solicitar uma lista de solicitações de criação de conta para uma organização que foram concluídas com êxito:

```
aws organizations list-create-account-status --states SUCCEEDED
```

A saída inclui uma matriz de objetos com informações sobre cada solicitação.

```

{
  "CreateAccountStatuses": [
    {
      "AccountId": "4444444444444444",
      "AccountName": "Developer Test Account",
      "CompletedTimeStamp": 1481835812.143,
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "State": "SUCCEEDED"
    }
  ]
}

```

Exemplo 2: Para recuperar uma lista das solicitações de criação de conta em andamento feitas na organização atual

O exemplo a seguir obtém uma lista de solicitações de criação de contas em andamento para uma organização:

```
aws organizations list-create-account-status --states IN_PROGRESS
```

A saída inclui uma matriz de objetos com informações sobre cada solicitação.

```
{
  "CreateAccountStatuses": [
    {
      "State": "IN_PROGRESS",
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "AccountName": "Production Account"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListCreateAccountStatus](#) na Referência de AWS CLI Comandos.

list-handshakes-for-account

O código de exemplo a seguir mostra como usar `list-handshakes-for-account`.

AWS CLI

Para recuperar uma lista dos apertos de mão enviados para uma conta

O exemplo a seguir mostra como obter uma lista de todos os apertos de mão associados à conta das credenciais usadas para chamar a operação:

```
aws organizations list-handshakes-for-account
```

A saída inclui uma lista de estruturas de aperto de mão com informações sobre cada aperto de mão, incluindo seu estado atual:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [ListHandshakesForAccount](#) na Referência de AWS CLI Comandos.

list-handshakes-for-organization

O código de exemplo a seguir mostra como usar `list-handshakes-for-organization`.

AWS CLI

Para recuperar uma lista dos apertos de mão associados a uma organização

O exemplo a seguir mostra como obter uma lista de apertos de mão associados à organização atual:

```
aws organizations list-handshakes-for-organization
```

A saída mostra dois apertos de mão. O primeiro é um convite para a conta de Juan e mostra um estado de OPEN. O segundo é um convite para a conta de Anika e mostra um estado de ACCEPTED:

```
{
  "Handshakes": [
    {
      "Action": "INVITE",
      "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
      "ExpirationTimestamp": 1482952459.257,
      "Id": "h-examplehandshakeid111",
      "Parties": [
        {
          "Id": "o-exampleorgid",
          "Type": "ORGANIZATION"
        },
        {
          "Id": "juan@example.com",
          "Type": "EMAIL"
        }
      ],
      "RequestedTimestamp": 1481656459.257,
      "Resources": [
        {
          "Resources": [
            {
              "Type": "MASTER_EMAIL",
              "Value": "bill@amazon.com"
            }
          ]
        }
      ]
    }
  ]
}
```

```

        {
            "Type": "MASTER_NAME",
            "Value": "Org Master
Account"
        },
        {
            "Type":
"ORGANIZATION_FEATURE_SET",
            "Value": "FULL"
        }
    ],
    "Type": "ORGANIZATION",
    "Value": "o-exampleorgid"
},
{
    "Type": "EMAIL",
    "Value": "juan@example.com"
},
{
    "Type": "NOTES",
    "Value": "This is an invitation to Juan's
account to join Bill's organization."
}
],
"State": "OPEN"
},
{
    "Action": "INVITE",
    "State": "ACCEPTED",
    "Arn": "arn:aws:organizations::111111111111:handshake/o-
exampleorgid/invite/h-examplehandshakeid111",
    "ExpirationTimestamp": 1.471797437427E9,
    "Id": "h-examplehandshakeid222",
    "Parties": [
        {
            "Id": "o-exampleorgid",
            "Type": "ORGANIZATION"
        },
        {
            "Id": "anika@example.com",
            "Type": "EMAIL"
        }
    ]
},
"RequestedTimestamp": 1.469205437427E9,

```

```

    "Resources": [
      {
        "Resources": [
          {
            "Type": "MASTER_EMAIL",
            "Value": "bill@example.com"
          },
          {
            "Type": "MASTER_NAME",
            "Value": "Master Account"
          }
        ],
        "Type": "ORGANIZATION",
        "Value": "o-exampleorgid"
      },
      {
        "Type": "EMAIL",
        "Value": "anika@example.com"
      },
      {
        "Type": "NOTES",
        "Value": "This is an invitation to Anika's
account to join Bill's organization."
      }
    ]
  }
]
}

```

- Para API obter detalhes, consulte [ListHandshakesForOrganization](#) na Referência de AWS CLI Comandos.

list-organizational-units-for-parent

O código de exemplo a seguir mostra como usar `list-organizational-units-for-parent`.

AWS CLI

Para recuperar uma lista de OUs em uma OU principal ou raiz

O exemplo a seguir mostra como obter uma lista de OUs em uma raiz especificada:

```
aws organizations list-organizational-units-for-parent --parent-id r-examplerootid111
```

A saída mostra que a raiz especificada contém duas OUs e mostra detalhes de cada uma:

```
{
  "OrganizationalUnits": [
    {
      "Name": "AccountingDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-examplerootid111/ou-examplerootid111-exampleoid111"
    },
    {
      "Name": "ProductionDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-examplerootid111/ou-examplerootid111-exampleoid222"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListOrganizationalUnitsForParent](#) na Referência de AWS CLI Comandos.

list-parents

O código de exemplo a seguir mostra como usar `list-parents`.

AWS CLI

Para listar o pai OUs ou as raízes de uma conta ou unidade organizacional secundária

O exemplo a seguir mostra como listar a OU raiz ou principal que contém essa conta 44444444444444:

```
aws organizations list-parents --child-id 44444444444444
```

A saída mostra que a conta especificada está na OU com o ID especificado:

```
{
  "Parents": [
```

```

    {
      "Id": "ou-examplerootid111-exampleouid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListParents](#) na Referência de AWS CLI Comandos.

list-policies-for-target

O código de exemplo a seguir mostra como usar `list-policies-for-target`.

AWS CLI

Para recuperar uma lista dos SCPs anexados diretamente a uma conta

O exemplo a seguir mostra como obter uma lista de todas as políticas de controle de serviço (SCPs), conforme especificado pelo parâmetro `Filter`, que estão diretamente vinculadas a uma conta:

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-id 444444444444
```

O resultado inclui uma lista de estruturas políticas com informações resumidas sobre as políticas. A lista não inclui políticas que se aplicam à conta devido à herança de sua localização em uma hierarquia de UO:

```

{
  "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."
    }
  ]
}

```

```
}
```

- Para API obter detalhes, consulte [ListPoliciesForTarget](#) na Referência de AWS CLI Comandos.

list-policies

O código de exemplo a seguir mostra como usar `list-policies`.

AWS CLI

Como recuperar uma lista de todas as políticas de um determinado tipo de uma organização

O exemplo a seguir mostra como obter uma lista de SCPs, conforme especificado pelo parâmetro `filter`:

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

A saída inclui uma lista de políticas com informações resumidas:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListPolicies](#) na Referência de AWS CLI Comandos.

list-roots

O código de exemplo a seguir mostra como usar `list-roots`.

AWS CLI

Para recuperar uma lista das raízes em uma organização

Este exemplo mostra como obter a lista de raízes de uma organização:

```
aws organizations list-roots
```

A saída inclui uma lista de estruturas raiz com informações resumidas:

```

{
  "Roots": [
    {
      "Name": "Root",
      "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
      "Id": "r-examplerootid111",
      "PolicyTypes": [
        {
          "Status": "ENABLED",
          "Type": "SERVICE_CONTROL_POLICY"
        }
      ]
    }
  ]
}

```



```
}

```

- Para API obter detalhes, consulte [ListRoots](#) na Referência de AWS CLI Comandos.

list-targets-for-policy

O código de exemplo a seguir mostra como usar `list-targets-for-policy`.

AWS CLI

Para recuperar uma lista das raízes, OUs, e contas às quais uma política está anexada

O exemplo a seguir mostra como obter uma lista das raízes, OUs, e das contas às quais a política especificada está anexada:

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

A saída inclui uma lista de objetos anexos com informações resumidas sobre as raízes e as contas às quais a política está anexada: 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"
    }
  ]
}
```

```
}  
    ]  
}
```

- Para API obter detalhes, consulte [ListTargetsForPolicy](#) na Referência de AWS CLI Comandos.

move-account

O código de exemplo a seguir mostra como usar `move-account`.

AWS CLI

Para mover uma conta entre raízes ou OUs

O exemplo a seguir mostra como mover a conta principal na organização da raiz para uma OU:

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-examplerootid111 --destination-parent-id ou-examplerootid111-exampleoid111
```

- Para API obter detalhes, consulte [MoveAccount](#) na Referência de AWS CLI Comandos.

remove-account-from-organization

O código de exemplo a seguir mostra como usar `remove-account-from-organization`.

AWS CLI

Para remover uma conta de uma organização como conta principal

O exemplo a seguir mostra como remover uma conta de uma organização:

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- Para API obter detalhes, consulte [RemoveAccountFromOrganization](#) na Referência de AWS CLI Comandos.

update-organizational-unit

O código de exemplo a seguir mostra como usar `update-organizational-unit`.

AWS CLI

Para renomear uma OU

Este exemplo mostra como renomear uma OU: Neste exemplo, a OU é renomeada para “AccountingOU”:

```
aws organizations update-organizational-unit --organizational-unit-id ou-examplerootid111-exampleoid111 --name AccountingOU
```

A saída mostra o novo nome:

```
{
  "OrganizationalUnit": {
    "Id": "ou-examplerootid111-exampleoid111"
    "Name": "AccountingOU",
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-examplerootid111-exampleoid111"
  }
}
```

- Para API obter detalhes, consulte [UpdateOrganizationalUnit](#) na Referência de AWS CLI Comandos.

update-policy

O código de exemplo a seguir mostra como usar update-policy.

AWS CLI

Exemplo 1: Para renomear uma política

O update-policy exemplo a seguir renomeia uma política e fornece uma nova descrição.

```
aws organizations update-policy \
  --policy-id p-examplepolicyid111 \
  --name Renamed-Policy \
  --description "This description replaces the original."
```

A saída mostra o novo nome e a descrição.

```
{
  "Policy": {
    "Content": "{\n  \"Version\": \"2012-10-17\", \n  \"Statement\": {\n\n    \"Effect\": \"Allow\", \n    \"Action\": \"ec2:*\", \n    \"Resource\": \"*\"\n  }\n}\n",
    "PolicySummary": {
      "Id": "p-examplepolicyid111",
      "AwsManaged": false,
      "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/service_control_policy/p-examplepolicyid111",
      "Description": "This description replaces the original.",
      "Name": "Renamed-Policy",
      "Type": "SERVICE_CONTROL_POLICY"
    }
  }
}
```

Exemplo 2: Para substituir o conteúdo de JSON texto de uma política

O exemplo a seguir mostra como substituir o JSON texto do SCP exemplo anterior por uma nova string de texto de JSON política que permite S3 em vez de EC2:

```
aws organizations update-policy \
  --policy-id p-examplepolicyid111 \
  --content '{"Version": "2012-10-17", "Statement": {"Effect": "Allow", "Action": "s3:*", "Resource": "*"}}
```

A saída mostra o novo conteúdo:

```
{
  "Policy": {
    "Content": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\": \"Allow\", \"Action\": \"s3:*\", \"Resource\": \"*\" } }",
    "PolicySummary": {
      "Arn": "arn:aws:organizations::111111111111:policy/o-exampleorgid/service_control_policy/p-examplepolicyid111",
      "AwsManaged": false;
      "Description": "This description replaces the original.",
      "Id": "p-examplepolicyid111",
      "Name": "Renamed-Policy",
      "Type": "SERVICE_CONTROL_POLICY"
    }
  }
}
```



```
{
  "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"
}
```

Para obter mais informações, consulte [Iniciar uma instância em seu Outpost no Guia](#) do usuário do AWS Outposts.

- Para API obter detalhes, consulte [GetOutpostInstanceTypes](#) na Referência de AWS CLI Comandos.

get-outpost

O código de exemplo a seguir mostra como usar `get-outpost`.

AWS CLI

Para obter detalhes do Outpost

O `get-outpost` exemplo a seguir exibe os detalhes do Posto Avançado especificado.

```
aws outposts get-outpost \
  --outpost-id op-0ab23c4567EXAMPLE
```

Saída:

```
{
  "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": {}
  }
}
```

Para obter mais informações, consulte [Trabalhando com Postos Avançados](#) no Guia do Usuário de Postos AWS Avançados.

- Para API obter detalhes, consulte [GetOutpost](#) na Referência de AWS CLI Comandos.

list-outposts

O código de exemplo a seguir mostra como usar `list-outposts`.

AWS CLI

Para listar Outposts

O `list-outposts` exemplo a seguir lista os Outposts em sua AWS conta.

```
aws outposts list-outposts
```

Saída:

```
{
  "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": {}
  }
]
```

Para obter mais informações, consulte [Trabalhando com Postos Avançados](#) no Guia do Usuário de Postos AWS Avançados.

- Para API obter detalhes, consulte [ListOutposts](#) na Referência de AWS CLI Comandos.

list-sites

O código de exemplo a seguir mostra como usar `list-sites`.

AWS CLI

Para listar sites

O `list-sites` exemplo a seguir lista os sites Outpost disponíveis em sua AWS conta.

```
aws outposts list-sites
```

Saída:

```
{
  "Sites": [
```



```
{
  "SiteId": "os-0ab12c3456EXAMPLE",
  "AccountId": "123456789012",
  "Name": "EXAMPLE",
  "Description": "example",
  "Tags": {}
}
]
```

Para obter mais informações, consulte [Trabalhando com Postos Avançados](#) no Guia do Usuário de Postos AWS Avançados.

- Para API obter detalhes, consulte [ListSites](#) na Referência de AWS CLI Comandos.

AWS Payment Cryptography exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Payment Cryptography.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-alias

O código de exemplo a seguir mostra como usar `create-alias`.

AWS CLI

Para criar um alias para uma chave

O `create-alias` exemplo a seguir cria um alias para uma chave.

```
aws payment-cryptography create-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifl1w2h
```

Saída:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaifl1w2h"  
  }  
}
```

Para obter mais informações, consulte [Sobre aliases](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [CreateAlias](#) na Referência de AWS CLI Comandos.

create-key

O código de exemplo a seguir mostra como usar create-key.

AWS CLI

Para criar uma chave

O create-key exemplo a seguir gera uma KEY TDES chave 2 que você pode usar para gerar e verificar CVV2 valores CVV /.

```
aws payment-cryptography create-key \  
  --exportable \  
  --key-  
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY, KeyClass=SYMMETRIC
```

Saída:

```
{  
  "Key": {
```

```

    "CreateTimestamp": "1686800690",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaiifllw2h",
    "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"
  }
}

```

Para obter mais informações, consulte [Geração de chaves](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [CreateKey](#) na Referência de AWS CLI Comandos.

delete-alias

O código de exemplo a seguir mostra como usar delete-alias.

AWS CLI

Para excluir um alias

O delete-alias exemplo a seguir exclui um alias. Isso não afeta a chave.

```
aws payment-cryptography delete-alias \  
  --alias-name alias/sampleAlias1
```

Este comando não produz saída.

Para obter mais informações, consulte [Sobre aliases](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [DeleteAlias](#) na Referência de AWS CLI Comandos.

delete-key

O código de exemplo a seguir mostra como usar delete-key.

AWS CLI

Para excluir uma chave

O delete-key exemplo a seguir agenda uma chave para exclusão após 7 dias, que é o período de espera padrão.

```
aws payment-cryptography delete-key \  
  --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/  
  kwapwa6qaifllw2h
```

Saída:

```
{  
  "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"
}
}

```

Para obter mais informações, consulte [Excluindo chaves](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [DeleteKey](#) na Referência de AWS CLI Comandos.

export-key

O código de exemplo a seguir mostra como usar `export-key`.

AWS CLI

Para exportar uma chave

O `export-key` exemplo a seguir exporta uma chave.

```

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 } }'

```

Conteúdo de `wrapping-key-certificate.pem`:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ01SQU1ZZS8xMXFUK2svVz1RUDJQ0E1V
```

Saída:

```
{
  "WrappedKey": {
    "KeyMaterial":
"308205A106092A864886F70D010702A08205923082058E020101310D300B06096086480165030402013082031F
    "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
  }
}
```

Para obter mais informações, consulte [Exportar chaves](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [ExportKey](#) na Referência de AWS CLI Comandos.

get-alias

O código de exemplo a seguir mostra como usar `get-alias`.

AWS CLI

Para obter um alias

O `get-alias` exemplo a seguir retorna ARN a chave associada ao alias.

```
aws payment-cryptography get-alias \
  --alias-name alias/sampleAlias1
```

Saída:

```
{
  "Alias": {
    "AliasName": "alias/sampleAlias1",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
  }
}
```

```
}
```

Para obter mais informações, consulte [Sobre aliases](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [GetAlias](#) na Referência de AWS CLI Comandos.

get-key

O código de exemplo a seguir mostra como usar get-key.

AWS CLI

Para obter os metadados de uma chave

O get-key exemplo a seguir retorna os metadados da chave associada ao alias. Essa operação não retorna material criptográfico.

```
aws payment-cryptography get-key \  
--key-identifier alias/sampleAlias1
```

Saída:

```
{  
  "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"
}
}

```

Para obter mais informações, consulte [Obter chaves](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [GetKey](#) na Referência de AWS CLI Comandos.

get-parameters-for-export

O código de exemplo a seguir mostra como usar `get-parameters-for-export`.

AWS CLI

Para inicializar o processo de exportação

O `get-parameters-for-export` exemplo a seguir gera um par de chaves, assina a chave e retorna o certificado e a raiz do certificado.

```

aws payment-cryptography get-parameters-for-export \
  --signing-key-algorithm RSA_2048 \
  --key-material-type TR34_KEY_BLOCK

```

Saída:

```

{
  "ExportToken": "export-token-ep5cwyzone7oya53",
  "ParametersValidUntilTimestamp": "1687415640",
  "SigningKeyAlgorithm": "RSA_2048",
  "SigningKeyCertificate":

  "MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC

```



```

VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
"SigningKeyCertificateChain":
"NIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1ZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}

```

Para obter mais informações, consulte [Exportar chaves](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [GetParametersForExport](#) na Referência de AWS CLI Comandos.

get-parameters-for-import

O código de exemplo a seguir mostra como usar `get-parameters-for-import`.

AWS CLI

Para inicializar o processo de importação

O `get-parameters-for-import` exemplo a seguir gera um par de chaves, assina a chave e retorna o certificado e a raiz do certificado.

```
aws payment-cryptography get-parameters-for-import \
  --key-material-type TR34_KEY_BLOCK \
  --wrapping-key-algorithm RSA_2048
```

Saída:

```
{
  "ImportToken": "import-token-qgmafpaa7nt2kfbb",
  "ParametersValidUntilTimestamp": "1687415640",
  "WrappingKeyAlgorithm": "RSA_2048",
  "WrappingKeyCertificate":
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySwTC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvJx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "WrappingKeyCertificateChain":
  "NIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySwTC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvJx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
```

```
}

```

Para obter mais informações, consulte [Importar chaves](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [GetParametersForImport](#) na Referência de AWS CLI Comandos.

get-public-key-certificate

O código de exemplo a seguir mostra como usar `get-public-key-certificate`.

AWS CLI

Para devolver a chave pública

O `get-public-key-certificate` exemplo a seguir retorna a parte da chave pública de um par de chaves.

```
aws payment-cryptography get-public-key-certificate \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

Saída:

```
{
  "KeyCertificate":
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVoQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHmAd
  BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VoQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHmAdBgkqhkiG9w0BCQEWEG5vb251QGFT
  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waL65M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1LJ00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "KeyCertificateChain":
```

```

"NIICiTCCAFICCCQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAdDgYDVQQHEwDTZWf0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAdDgYD
VQQHEwDTZWf0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}

```

Para obter mais informações, consulte [Obter a chave pública/certificado associado a um par de chaves no Guia](#) do usuário da criptografia AWS de pagamento.

- Para API obter detalhes, consulte [GetPublicKeyCertificate](#) na Referência de AWS CLI Comandos.

import-key

O código de exemplo a seguir mostra como usar `import-key`.

AWS CLI

Para importar uma chave TR-34

O `import-key` exemplo a seguir importa uma chave TR-34.

```

aws payment-cryptography import-key \
  --key-material='{ "Tr34KeyBlock": {" \
    CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564nijnm", \
    "ImportToken": "import-token-5ott6ho5nts7bbc", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "SigningKeyCertificate": file://signing-key-certificate.pem, \
    "WrappedKeyBlock": file://wrapped-key-block.pem } }'

```

Conteúdo de `signing-key-certificate.pem`:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ01RYWVCK25IbE1WZU1PR1ZiNjU1Q2Zj
```

Conteúdo de wrapped-key-block.pem:

```
3082059806092A864886F70D010702A082058930820585020101310D300B06096086480165030402013082031606
```

Saída:

```
{
  "Key": {
    "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
    bzmvgyx dg3skt wxd",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_2KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": false,
        "DeriveKey": false,
        "Encrypt": false,
        "Generate": true,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": false,
        "Verify": true,
        "Wrap": false
      },
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
    },
    "KeyCheckValue": "D9B20E",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "EXTERNAL",
    "KeyState": "CREATE_COMPLETE",
    "UsageStartTimestamp": "2023-06-09T16:56:27.621000-07:00"
  }
}
```

Para obter mais informações, consulte [Importar chaves](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [ImportKey](#) na Referência de AWS CLI Comandos.

list-aliases

O código de exemplo a seguir mostra como usar `list-aliases`.

AWS CLI

Para obter uma lista de aliases

O `list-aliases` exemplo a seguir mostra todos os aliases da sua conta nessa região.

```
aws payment-cryptography list-aliases
```

Saída:

```
{
  "Aliases": [
    {
      "AliasName": "alias/sampleAlias1",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/kwapwa6qaiif1lw2h"
    },
    {
      "AliasName": "alias/sampleAlias2",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/kwapwa6qaiif1lw2h"
    }
  ]
}
```

Para obter mais informações, consulte [Sobre aliases](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [ListAliases](#) na Referência de AWS CLI Comandos.

list-keys

O código de exemplo a seguir mostra como usar `list-keys`.

AWS CLI

Para obter uma lista de chaves

O `list-keys` exemplo a seguir mostra todas as chaves da sua conta nessa região.

```
aws payment-cryptography list-keys
```

Saída:

```
{
  "Keys": [
    {
      "CreateTimestamp": "1666506840",
      "Enabled": false,
      "Exportable": true,
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
      kwapwa6qaiifllw2h",
      "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"
    }
  ]
}
```

Para obter mais informações, consulte [Listar chaves](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [ListKeys](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para obter a lista de tags de uma chave

O `list-tags-for-resource` exemplo a seguir obtém as tags de uma chave.

```
aws payment-cryptography list-tags-for-resource \
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlLw2h
```

Saída:

```
{
  "Tags": [
    {
      "Key": "BIN",
      "Value": "20151120"
    },
    {
      "Key": "Project",
      "Value": "Production"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de etiquetas-chave com API operações](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

restore-key

O código de exemplo a seguir mostra como usar `restore-key`.

AWS CLI

Para restaurar uma chave que está programada para exclusão

O `restore-key` exemplo a seguir cancela a exclusão de uma chave.

```
aws payment-cryptography restore-key \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifl1w2h
```

Saída:

```
{  
  "Key": {  
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaifl1w2h",  
    "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"  
  }  
}
```

Para obter mais informações, consulte [Excluindo chaves](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [RestoreKey](#) na Referência de AWS CLI Comandos.

start-key-usage

O código de exemplo a seguir mostra como usar `start-key-usage`.

AWS CLI

Para habilitar uma chave

O `start-key-usage` exemplo a seguir permite que uma chave seja usada.

```
aws payment-cryptography start-key-usage \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaif1lw2h
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Ativação e desativação de chaves](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [StartKeyUsagena Referência de AWS CLI Comandos](#).

stop-key-usage

O código de exemplo a seguir mostra como usar stop-key-usage.

AWS CLI

Para desativar uma chave

O stop-key-usage exemplo a seguir desativa uma chave.

```

aws payment-cryptography stop-key-usage \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaif1lw2h

```

Saída:

```

{
  "Key": {
    "CreateTimestamp": "1686800690",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/alsuwxug3pgy6xh",
    "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"
}
}

```

Para obter mais informações, consulte [Ativação e desativação de chaves](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [StopKeyUsage](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar uma chave

O tag-resource exemplo a seguir marca uma chave.

```

aws payment-cryptography tag-resource \
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlw2h \
  --tags Key=sampleTag,Value=sampleValue

```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags de chave](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para remover uma tag de uma chave

O `untag-resource` exemplo a seguir remove uma tag de uma chave.

```
aws payment-cryptography untag-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlw2h \  
  --tag-keys sampleTag
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de tags de chave](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-alias

O código de exemplo a seguir mostra como usar `update-alias`.

AWS CLI

Para atualizar um alias

O `update-alias` exemplo a seguir associa o alias a uma chave diferente.

```
aws payment-cryptography update-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  tqv5yij6wtxx64pi
```

Saída:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
    tqv5yij6wtxx64pi "  
  }  
}
```

Para obter mais informações, consulte [Sobre aliases](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [UpdateAlias](#) na Referência de AWS CLI Comandos.

AWS Payment Cryptography Exemplos de planos de dados usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com AWS Payment Cryptography Data Plane.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

decrypt-data

O código de exemplo a seguir mostra como usar `decrypt-data`.

AWS CLI

Para decifrar texto cifrado

O `decrypt-data` exemplo a seguir decodifica dados de texto cifrado usando uma chave simétrica. Para essa operação, a chave deve ter sido `KeyModesOfUse` definida como `Decrypt` e `KeyUsage` definida como `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}'
```

Saída:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiifllw2h",
  "KeyCheckValue": "71D7AE",
  "PlainText": "31323334313233343132333431323334"
}
```

Para obter mais informações, consulte [Descriptografar dados](#) no Guia do usuário de criptografia AWS de pagamento.

- Para API obter detalhes, consulte [DecryptData](#) na Referência de AWS CLI Comandos.

encrypt-data

O código de exemplo a seguir mostra como usar `encrypt-data`.

AWS CLI

Para criptografar dados

O `encrypt-data` exemplo a seguir criptografa dados em texto simples usando uma chave simétrica. Para essa operação, a chave deve ter sido `KeyModesOfUse` definida como `Encrypt` e `KeyUsage` definida como `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`.

```
aws payment-cryptography-data encrypt-data \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiifllw2h \
  --plain-text 31323334313233343132333431323334 \
  --encryption-attributes 'Symmetric={Mode=CBC}'
```

Saída:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiifllw2h",
  "KeyCheckValue": "71D7AE",
  "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

```
}
```

Para obter mais informações, consulte [Criptografar dados](#) no Guia do usuário AWS de criptografia de pagamento.

- Para API obter detalhes, consulte [EncryptData](#) na Referência de AWS CLI Comandos.

generate-card-validation-data

O código de exemplo a seguir mostra como usar `generate-card-validation-data`.

AWS CLI

Para gerar um CVV

O `generate-card-validation-data` exemplo a seguir gera um 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}
```

Saída:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
  "KeyCheckValue": "CADD1",
  "ValidationData": "801"
}
```

Para obter mais informações, consulte [Gerar dados do cartão](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [GenerateCardValidationData](#) na Referência de AWS CLI Comandos.

generate-mac

O código de exemplo a seguir mostra como usar `generate-mac`.

AWS CLI

Para gerar um MAC

O `generate-card-validation-data` exemplo a seguir gera um Código de Autenticação de Mensagem Baseado em Hash (HMAC) para autenticação de dados do cartão usando o algoritmo HMAC_SHA256 e uma chave de HMAC criptografia. A chave deve estar KeyUsage configurada como TR31_M7_HMAC_KEY e KeyModesOfUse paraGenerate.

```
aws payment-cryptography-data generate-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaiFlw2h \  
  --message-  
data "3b313038383439303031303733393431353d32343038323236303030373030303f33" \  
  --generation-attributes Algorithm=HMAC_SHA256
```

Saída:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaiFlw2h,  
  "KeyCheckValue": "2976E7",  
  "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"  
}
```

Para obter mais informações, consulte [Generate MAC](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [GenerateMac](#) na Referência de AWS CLI Comandos.

generate-pin-data

O código de exemplo a seguir mostra como usar `generate-pin-data`.

AWS CLI

Para gerar um PIN

O `generate-card-validation-data` exemplo a seguir gera um novo aleatório PIN usando o PIN esquema 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}

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerar PIN dados](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [GeneratePinData](#) na Referência de AWS CLI Comandos.

re-encrypt-data

O código de exemplo a seguir mostra como usar `re-encrypt-data`.

AWS CLI

Para criptografar novamente os dados com uma chave diferente

O `re-encrypt-data` exemplo a seguir descriptografa o texto cifrado que foi criptografado usando uma chave AES simétrica e o recriptografa usando uma chave exclusiva derivada por transação (). 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=0123456789111111}" \
--outgoing-encryption-attributes '{"Symmetric": {"Mode": "ECB"}}'

```

Saída:

```

{
  "CipherText":
    "F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
  "KeyCheckValue": "2E8CD9"
}

```

Para obter mais informações, consulte [Criptografar e descriptografar dados](#) no Guia do usuário de criptografia de AWS pagamento.

- Para API obter detalhes, consulte [ReEncryptData](#) na Referência de AWS CLI Comandos.

translate-pin-data

O código de exemplo a seguir mostra como usar translate-pin-data.

AWS CLI

Para traduzir PIN dados

O translate-pin-data exemplo a seguir traduz a PEK TDES criptografia PIN de um usando o PIN bloco ISO 0 para um PIN bloco AES ISO 4 usando o 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"

```

Saída:

```

{
  "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/ivi5ksfsuplneuyt",
  "KeyCheckValue": "7CC9E2"
}

```

Para obter mais informações, consulte [Translate PIN data](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [TranslatePinData](#) na Referência de AWS CLI Comandos.

verify-auth-request-cryptogram

O código de exemplo a seguir mostra como usar `verify-auth-request-cryptogram`.

AWS CLI

Para verificar uma solicitação de autenticação

O `verify-auth-request-cryptogram` exemplo a seguir verifica um criptograma de solicitação de autorização (ARQC).

```

aws payment-cryptography-data verify-auth-request-cryptogram \
--auth-request-cryptogram FGE1BD1E6037FB3E \
--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"
```

Saída:

```
{
  "AuthResponseValue": "D899B8C6FBF971AA",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya",
  "KeyCheckValue": "985792"
}
```

Para obter mais informações, consulte [Verify auth request \(ARQC\) criptograma no Guia do usuário AWS de criptografia de pagamento](#).

- Para API obter detalhes, consulte [VerifyAuthRequestCryptogram](#) na Referência de AWS CLI Comandos.

verify-card-validation-data

O código de exemplo a seguir mostra como usar `verify-card-validation-data`.

AWS CLI

Para validar um CVV

O `verify-card-validation-data` exemplo a seguir valida uma CVV/CVV2 para 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
```

Saída:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
  "KeyCheckValue": "CADD1"
}
```

Para obter mais informações, consulte [Verificar dados do cartão](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [VerifyCardValidationData](#) na Referência de AWS CLI Comandos.

verify-mac

O código de exemplo a seguir mostra como usar `verify-mac`.

AWS CLI

Para verificar um MAC

O `verify-mac` exemplo a seguir verifica um Código de Autenticação de Mensagem Baseado em Hash (HMAC) para autenticação de dados do cartão usando o algoritmo HMAC _ SHA256 e uma chave de HMAC criptografia.

```
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 ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

Saída:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qnobl5lghrzunce6",  
  "KeyCheckValue": "2976E7",  
}
```

Para obter mais informações, consulte [Verificar MAC](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [VerifyMac](#) na Referência de AWS CLI Comandos.

verify-pin-data

O código de exemplo a seguir mostra como usar `verify-pin-data`.

AWS CLI

Para verificar um PIN

O `verify-pin-data` exemplo a seguir valida a PIN para a PAN

```
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
```

Saída:

```
{
  "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",
}
```

Para obter mais informações, consulte [Verificar PIN dados](#) no Guia do usuário da criptografia de AWS pagamento.

- Para API obter detalhes, consulte [VerifyPinData](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Pinpoint usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Pinpoint.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-app

O código de exemplo a seguir mostra como usar `create-app`.

AWS CLI

Exemplo 1: como criar uma aplicação

O exemplo de `create-app` a seguir cria uma aplicação (projeto).

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp
```

Saída:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {}  
  }  
}
```

Exemplo 2: como criar uma aplicação marcada

O exemplo de `create-app` a seguir cria uma aplicação (projeto) e associa uma tag (chave e valor) a ela.

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```


Saída:

```
{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Stack": "Test"
    }
  }
}
```

- Para API obter detalhes, consulte [CreateAppna](#) Referência de AWS CLI Comandos.

create-sms-template

O código de exemplo a seguir mostra como usar `create-sms-template`.

AWS CLI

Cria um modelo de mensagem para mensagens enviadas pelo SMS canal

O `create-sms-template` exemplo a seguir cria um modelo de SMS mensagem.

```
aws pinpoint create-sms-template \
  --template-name TestTemplate \
  --sms-template-request file://myfile.json \
  --region us-east-1
```

Conteúdo de `myfile.json`:

```
{
  "Body": "hello\n how are you?\n food is good",
  "TemplateDescription": "Test SMS Template"
}
```

Saída:

```
{
  "CreateTemplateMessageBody": {
```

```

    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
    "Message": "Created",
    "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"
  }
}

```

Para obter mais informações, consulte os [modelos de mensagem do Amazon Pinpoint no Guia](#) do usuário do Amazon Pinpoint.

- Para API obter detalhes, consulte [CreateSmsTemplate](#) na Referência de AWS CLI Comandos.

delete-app

O código de exemplo a seguir mostra como usar `delete-app`.

AWS CLI

Como excluir uma aplicação

O exemplo de `delete-app` a seguir exclui uma aplicação (projeto).

```

aws pinpoint delete-app \
  --application-id 810c7aab86d42fb2b56c8c966example

```

Saída:

```

{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {}
  }
}

```

- Para API obter detalhes, consulte [DeleteApp](#) na Referência de AWS CLI Comandos.

get-apns-channel

O código de exemplo a seguir mostra como usar `get-apns-channel`.

AWS CLI

Para recuperar informações sobre o status e as configurações do APNs canal para um aplicativo

O `get-apns-channel` exemplo a seguir recupera informações sobre o status e as configurações do APNs canal para um aplicativo.

```
aws pinpoint get-apns-channel \  
  --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \  
  --region us-east-1
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [GetApnsChannel](#) na Referência de AWS CLI Comandos.

get-app

O código de exemplo a seguir mostra como usar `get-app`.

AWS CLI

Para recuperar informações sobre um aplicativo (projeto)

O `get-app` exemplo a seguir recupera informações sobre um aplicativo (projeto).

```
aws pinpoint get-app \  
  --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \  
  --region us-east-1
```

```
--application-id 810c7aab86d42fb2b56c8c966example \  
--region us-east-1
```

Saída:

```
{  
  "ApplicationResponse": {  
    "Arn": "arn:aws:mobiletargeting:us-  
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",  
    "Id": "810c7aab86d42fb2b56c8c966example",  
    "Name": "ExampleCorp",  
    "tags": {  
      "Year": "2019",  
      "Stack": "Production"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetApp](#)na Referência de AWS CLI Comandos.

get-apps

O código de exemplo a seguir mostra como usar get-apps.

AWS CLI

Para recuperar informações sobre todos os seus aplicativos

O get-apps exemplo a seguir recupera informações sobre todos os seus aplicativos (projetos).

```
aws pinpoint get-apps
```

Saída:

```
{  
  "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"
}
}

```

A presença do valor da `NextToken` resposta indica que há mais saída disponível. Chame o comando novamente e forneça esse valor como parâmetro `NextToken` de entrada.

- Para API obter detalhes, consulte [GetApps](#) na Referência de AWS CLI Comandos.

get-campaign

O código de exemplo a seguir mostra como usar `get-campaign`.

AWS CLI

Para recuperar informações sobre o status, a configuração e outras configurações de uma campanha

O `get-campaign` exemplo a seguir recupera informações sobre o status, a configuração e outras configurações de uma campanha.

```
aws pinpoint get-campaign \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \  
  --region us-east-1
```

Saída:

```
{  
  "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
  }
}

```

- Para API obter detalhes, consulte [GetCampaign](#) na Referência de AWS CLI Comandos.

get-campaigns

O código de exemplo a seguir mostra como usar `get-campaigns`.

AWS CLI

Para recuperar informações sobre o status, a configuração e outras configurações de todas as campanhas associadas a um aplicativo

O `get-campaigns` exemplo a seguir recupera informações sobre o status, a configuração e outras configurações de todas as campanhas associadas a um aplicativo.

```

aws pinpoint get-campaigns \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1

```

Saída:

```

{
  "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  <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">\n    <head>\n    <meta http-equiv=\n"Content-Type\n" content=\n"text/html;\n
charset=utf-8\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
}
]
}
}

```

- Para API obter detalhes, consulte [GetCampaigns](#) na Referência de AWS CLI Comandos.

get-channels

O código de exemplo a seguir mostra como usar `get-channels`.

AWS CLI

Para recuperar informações sobre o histórico e o status de cada canal de um aplicativo

O `get-channels` exemplo a seguir recupera informações sobre o histórico e o status de cada canal de um aplicativo.

```
aws pinpoint get-channels \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Saída:

```
{  
  "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
  }
}
}
}

```

- Para API obter detalhes, consulte [GetChannels](#) na Referência de AWS CLI Comandos.

get-email-channel

O código de exemplo a seguir mostra como usar `get-email-channel`.

AWS CLI

Para recuperar informações sobre o status e as configurações do canal de e-mail de um aplicativo

O `get-email-channel` exemplo a seguir recupera o status e as configurações do canal de e-mail de um aplicativo.

```

aws pinpoint get-email-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1

```

Saída:

```

{
  "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
  }
}

```

```
}
```

- Para API obter detalhes, consulte [GetEmailChannel](#) na Referência de AWS CLI Comandos.

get-endpoint

O código de exemplo a seguir mostra como usar `get-endpoint`.

AWS CLI

Como recuperar informações sobre as configurações e os atributos de um endpoint específico de uma aplicação

O exemplo de `get-endpoint` a seguir recupera informações sobre as configurações e os atributos de um endpoint específico de uma aplicação.

```
aws pinpoint get-endpoint \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --endpoint-id testendpoint \  
  --region us-east-1
```

Saída:

```
{  
  "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"
    }
  }
}
```

- Para API obter detalhes, consulte [GetEndpoint](#) na Referência de AWS CLI Comandos.

get-gcm-channel

O código de exemplo a seguir mostra como usar `get-gcm-channel`.

AWS CLI

Para recuperar informações sobre o status e as configurações do GCM canal para um aplicativo

O `get-gcm-channel` exemplo a seguir recupera informações sobre o status e as configurações do GCM canal para um aplicativo.

```
aws pinpoint get-gcm-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

Saída:

```
{
  "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
  }
}
```

```
}
```

- Para API obter detalhes, consulte [GetGcmChannel](#) na Referência de AWS CLI Comandos.

get-sms-channel

O código de exemplo a seguir mostra como usar `get-sms-channel`.

AWS CLI

Para recuperar informações sobre o status e as configurações do SMS canal para um aplicativo

O exemplo de `get-sms-channel` a seguir recupera o status e as configurações do canal de SMS de uma aplicação.

```
aws pinpoint get-sms-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [GetSmsChannel](#) na Referência de AWS CLI Comandos.

get-sms-template

O código de exemplo a seguir mostra como usar `get-sms-template`.

AWS CLI

Recupera o conteúdo e as configurações de um modelo de mensagem para mensagens enviadas pelo canal SMS

O `get-sms-template` exemplo a seguir recupera o conteúdo e as configurações de um modelo de SMS mensagem.

```
aws pinpoint get-sms-template \  
  --template-name TestTemplate \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte os [modelos de mensagem do Amazon Pinpoint no Guia](#) do usuário do Amazon Pinpoint.

- Para API obter detalhes, consulte [GetSmsTemplate](#) na Referência de AWS CLI Comandos.

`get-voice-channel`

O código de exemplo a seguir mostra como usar `get-voice-channel`.

AWS CLI

Para recuperar informações sobre o status e as configurações do canal de voz de um aplicativo

O `get-voice-channel` exemplo a seguir recupera o status e as configurações do canal de voz de um aplicativo.

```
aws pinpoint get-voice-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [GetVoiceChannel](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para recuperar uma lista de tags para um recurso

O `list-tags-for-resource` exemplo a seguir recupera todas as tags (nomes e valores de chave) associadas ao recurso especificado.

```
aws pinpoint list-tags-for-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

Saída:


```
{
  "TagsModel": {
    "tags": {
      "Year": "2019",
      "Stack": "Production"
    }
  }
}
```

Para obter mais informações, consulte “Marcar recursos do Amazon Pinpoint <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>” no Guia do desenvolvedor do Amazon Pinpoint.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

phone-number-validate

O código de exemplo a seguir mostra como usar `phone-number-validate`.

AWS CLI

Recupera informações sobre um número de telefone

O seguinte `phone-number-validate` recupera informações sobre um número de telefone.

```
aws pinpoint phone-number-validate \
  --number-validate-request PhoneNumber="+12065550142" \
  --region us-east-1
```

Saída:

```
{
  "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"  
  }  
}
```

Para obter mais informações, consulte o [SMS canal Amazon Pinpoint no Guia](#) do usuário do Amazon Pinpoint.

- Para API obter detalhes, consulte [PhoneNumberValidate](#) na Referência de AWS CLI Comandos.

send-messages

O código de exemplo a seguir mostra como usar send-messages.

AWS CLI

Para enviar SMS mensagens usando o endpoint de um aplicativo

O exemplo de send-messages a seguir envia uma mensagem direta a uma aplicação com um endpoint.

```
aws pinpoint send-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --message-request file://myfile.json \  
  --region us-west-2
```

Conteúdo de myfile.json:

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Endpoints": {  
    "testendpoint": {}  
  }  
}
```

Saída:

```
{
  "MessageResponse": {
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
    "EndpointResult": {
      "testendpoint": {
        "Address": "+12345678900",
        "DeliveryStatus": "SUCCESSFUL",
        "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
        "StatusCode": 200,
        "StatusMessage": "MessageId:
itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
      }
    },
    "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
  }
}
```

Para obter mais informações, consulte o [SMS canal Amazon Pinpoint no Guia](#) do usuário do Amazon Pinpoint.

- Para API obter detalhes, consulte [SendMessages](#) na Referência de AWS CLI Comandos.

send-users-messages

O código de exemplo a seguir mostra como usar send-users-messages.

AWS CLI

Para enviar SMS mensagem para um usuário de um aplicativo

O send-users-messages exemplo a seguir envia uma mensagem direta para um usuário de um aplicativo.

```
aws pinpoint send-users-messages \
  --application-id 611e3e3cdd47474c9c1399a505665b91 \
  --send-users-message-request file://myfile.json \
  --region us-west-2
```

Conteúdo de myfile.json:

```
{
```

```

    "MessageConfiguration": {
      "SMSMessage": {
        "Body": "hello, how are you?"
      }
    },
    "Users": {
      "testuser": {}
    }
  }
}

```

Saída:

```

{
  "SendUsersMessageResponse": {
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
    "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",
    "Result": {
      "testuser": {
        "testuserendpoint": {
          "DeliveryStatus": "SUCCESSFUL",
          "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",
          "StatusCode": 200,
          "StatusMessage": "MessageId:
7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",
          "Address": "+12345678900"
        }
      }
    }
  }
}

```

Para obter mais informações, consulte o [SMS canal Amazon Pinpoint no Guia](#) do usuário do Amazon Pinpoint.

- Para API obter detalhes, consulte [SendUsersMessages](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um recurso

O exemplo a seguir adiciona duas tags (nomes e valores de chave) a um recurso.

```
aws pinpoint list-tags-for-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tags-model tags={Stack=Production,Year=2019}
```

Este comando não produz saída.

Para obter mais informações, consulte “Marcar recursos do Amazon Pinpoint <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>” no Guia do desenvolvedor do Amazon Pinpoint.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Exemplo 1: Para remover uma tag de um recurso

O `untag-resource` exemplo a seguir remove a tag especificada (nome e valor da chave) de um recurso.

```
aws pinpoint untag-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tag-keys Year
```

Este comando não produz saída.

Exemplo 2: Para remover várias tags de um recurso

O `untag-resource` exemplo a seguir remove as tags especificadas (nomes e valores de chave) de um recurso.

```
aws pinpoint untag-resource \  
  --resource-arn arn:aws:mobiletargeting:us-  
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tag-keys Year
```

```
--tag-keys Year Stack
```

Este comando não produz saída.

Para obter mais informações, consulte “Marcar recursos do Amazon Pinpoint [https://docs.aws.amazon.com/pinpoint/< latest/developerguide/tagging-resources.html>](https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html)’__ no Guia do desenvolvedor do Amazon Pinpoint.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-sms-channel

O código de exemplo a seguir mostra como usar `update-sms-channel`.

AWS CLI

Para habilitar o SMS canal ou atualizar o status e as configurações do SMS canal para um aplicativo.

O `update-sms-channel` exemplo a seguir ativa o SMS canal para um SMS canal para um aplicativo.

```
aws pinpoint update-sms-channel \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --sms-channel-request Enabled=true \  
  --region us-west-2
```

Saída:

```
{  
  "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  
  }  
}
```

```
}
```

Para obter mais informações, consulte o [SMS canal Amazon Pinpoint no Guia](#) do usuário do Amazon Pinpoint.

- Para API obter detalhes, consulte [UpdateSmsChannel](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Polly usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Polly.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-lexicon

O código de exemplo a seguir mostra como usar `delete-lexicon`.

AWS CLI

Como excluir um léxico

O exemplo de `delete-lexicon` a seguir exclui o léxico especificado.

```
aws polly delete-lexicon \  
  --name w3c
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar a DeleteLexicon operação](#) no Amazon Polly Developer Guide.

- Para API obter detalhes, consulte [DeleteLexicon](#) na Referência de AWS CLI Comandos.

get-lexicon

O código de exemplo a seguir mostra como usar `get-lexicon`.

AWS CLI

Como recuperar o conteúdo de um léxico

O exemplo de `get-lexicon` a seguir recupera o conteúdo do léxico de pronúncia especificado.

```
aws polly get-lexicon \  
  --name w3c
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte Como [usar a GetLexicon operação](#) no Amazon Polly Developer Guide.

- Para API obter detalhes, consulte [GetLexicon](#) na Referência de AWS CLI Comandos.

get-speech-synthesis-task

O código de exemplo a seguir mostra como usar `get-speech-synthesis-task`.

AWS CLI

Como obter informações sobre uma tarefa de síntese de fala

O exemplo de `get-speech-synthesis-task` a seguir recupera informações sobre a tarefa de síntese de fala especificada.

```
aws polly get-speech-synthesis-task \  
  --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criar arquivos de áudio longos](#) no Guia do desenvolvedor do Amazon Polly.

- Para API obter detalhes, consulte [GetSpeechSynthesisTask](#) na Referência de AWS CLI Comandos.

list-lexicons

O código de exemplo a seguir mostra como usar `list-lexicons`.

AWS CLI

Como listar os léxicos

O exemplo de `list-lexicons` a seguir lista os léxicos de pronúncia.

```
aws polly list-lexicons
```

Saída:

```
{
  "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
      }
    }
  ]
}
```

Para obter mais informações, consulte Como [usar a ListLexicons operação](#) no Amazon Polly Developer Guide.

- Para API obter detalhes, consulte [ListLexicons](#) na Referência de AWS CLI Comandos.

list-speech-synthesis-tasks

O código de exemplo a seguir mostra como usar `list-speech-synthesis-tasks`.

AWS CLI

Para listar suas tarefas de síntese de fala

O `list-speech-synthesis-tasks` exemplo a seguir lista suas tarefas de síntese de fala.

```
aws polly list-speech-synthesis-tasks
```

Saída:

```
{
```

```

"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"
  }
]
}

```

Para obter mais informações, consulte [Criar arquivos de áudio longos](#) no Guia do desenvolvedor do Amazon Polly.

- Para API obter detalhes, consulte [ListSpeechSynthesisTasks](#) na Referência de AWS CLI Comandos.

put-lexicon

O código de exemplo a seguir mostra como usar `put-lexicon`.

AWS CLI

Como armazenar um léxico

O exemplo de `put-lexicon` a seguir armazena o léxico de pronúncia especificado. O `example.pls` arquivo especifica um léxico compatível com o W3CPLS.

```

aws polly put-lexicon \
  --name w3c \
  --content file://example.pls

```

Conteúdo de `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>
}
```

Este comando não produz saída.

Para obter mais informações, consulte Como [usar a PutLexicon operação](#) no Amazon Polly Developer Guide.

- Para API obter detalhes, consulte [PutLexicon](#) na Referência de AWS CLI Comandos.

start-speech-synthesis-task

O código de exemplo a seguir mostra como usar start-speech-synthesis-task.

AWS CLI

Como sintetizar texto

O start-speech-synthesis-task exemplo a seguir sintetiza o texto `text_file.txt` e armazena o MP3 arquivo resultante no bucket especificado.

```
aws polly start-speech-synthesis-task \
  --output-format mp3 \
  --output-s3-bucket-name my-s3-bucket \
  --text file://text_file.txt \
  --voice-id Joanna
```

Saída:

```
{
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criar arquivos de áudio longos](#) no Guia do desenvolvedor do Amazon Polly.

- Para API obter detalhes, consulte [StartSpeechSynthesisTask](#) na Referência de AWS CLI Comandos.

AWS Price List exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Price List.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-services

O código de exemplo a seguir mostra como usar `describe-services`.

AWS CLI

Para recuperar metadados do serviço

Este exemplo recupera os metadados do código de EC2 serviço da Amazon.

Comando:

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --  
max-items 1
```

Saída:

```
{  
  "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"
}

```

- Para API obter detalhes, consulte [DescribeServices](#) na Referência de AWS CLI Comandos.

get-attribute-values

O código de exemplo a seguir mostra como usar get-attribute-values.

AWS CLI

Para recuperar uma lista de valores de atributos

O `get-attribute-values` exemplo a seguir recupera uma lista de valores disponíveis para o atributo especificado.

```
aws pricing get-attribute-values \  
  --service-code AmazonEC2 \  
  --attribute-name volumeType \  
  --max-items 2
```

Saída:

```
{  
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",  
  "AttributeValues": [  
    {  
      "Value": "Cold HDD"  
    },  
    {  
      "Value": "General Purpose"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [GetAttributeValues](#) na Referência de AWS CLI Comandos.

get-products

O código de exemplo a seguir mostra como usar `get-products`.

AWS CLI

Para recuperar uma lista de produtos

Este exemplo recupera uma lista de produtos que correspondem aos critérios fornecidos.

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"
  }
]
```

Saída:

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVSagyIFn770XQ0zdUIc09BY6ucBG9itXAZGZF/
zioUz0sUKh6PCcPWa0yPZRiMePb986TeoKYB9l55fw/
CyoMq5ymnGmT1Vj39Tljbba1hcqnVfTmPIilx8Uy5bdDaBYy/e/20fw9Edzsykbs8LTBUbNbiDQ
+BBds5yeI9AQkUepuKk3aEahFPxJ55kx/zk",
  "PriceList": [
    {
      "\product\":{\productFamily\":\Storage\", \attributes\":{\storageMedia\":
      \SSD-backed\", \maxThroughputvolume\":\320 MB/sec\", \volumeType\":\Provisioned
      IOPS\", \maxIopsvolume\":\20000\", \servicecode\":\AmazonEC2\", \usagetype
      \\":\APS1-EBS:VolumeUsage.piops\", \locationType\":\AWS Region\", \location\":
      \Asia Pacific (Singapore)\", \servicename\":\Amazon Elastic Compute Cloud\",
      \maxVolumeSize\":\16 TiB\", \operation\":\\", \sku\":\3MKHN58N7RDDVGKJ\",
      \serviceCode\":\AmazonEC2\", \terms\":{\OnDemand\":{\3MKHN58N7RDDVGKJ.JRTCKXETXF
      \\":{\priceDimensions\":{\3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\":{\unit\":\GB-
      Mo\", \endRange\":\Inf\", \description\":\$.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\"}
    ]
  }
}
```

- Para API obter detalhes, consulte [GetProducts](#) na Referência de AWS CLI Comandos.

AWS Private CA exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Private CA.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-certificate-authority-audit-report

O código de exemplo a seguir mostra como usar `create-certificate-authority-audit-report`.

AWS CLI

Para criar um relatório de auditoria da autoridade de certificação

O `create-certificate-authority-audit-report` comando a seguir cria um relatório de auditoria para a CA privada identificada peloARN.

```
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
```

- Para API obter detalhes, consulte [CreateCertificateAuthorityAuditReport](#) na Referência de AWS CLI Comandos.

create-certificate-authority

O código de exemplo a seguir mostra como usar `create-certificate-authority`.

AWS CLI

Para criar uma autoridade de certificação privada

O `create-certificate-authority` comando a seguir cria uma autoridade de certificação privada em sua AWS conta.

```
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
```

- Para API obter detalhes, consulte [CreateCertificateAuthority](#) na Referência de AWS CLI Comandos.

`delete-certificate-authority`

O código de exemplo a seguir mostra como usar `delete-certificate-authority`.

AWS CLI

Para excluir uma autoridade de certificação privada

O `delete-certificate-authority` comando a seguir exclui a autoridade de certificação identificada pelo 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
```

- Para API obter detalhes, consulte [DeleteCertificateAuthority](#) na Referência de AWS CLI Comandos.

`describe-certificate-authority-audit-report`

O código de exemplo a seguir mostra como usar `describe-certificate-authority-audit-report`.

AWS CLI

Para descrever um relatório de auditoria para uma autoridade de certificação

O `describe-certificate-authority-audit-report` comando a seguir lista informações sobre o relatório de auditoria especificado para a CA identificado peloARN.

```
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
```

- Para API obter detalhes, consulte [DescribeCertificateAuthorityAuditReport](#) na Referência de AWS CLI Comandos.

describe-certificate-authority

O código de exemplo a seguir mostra como usar `describe-certificate-authority`.

AWS CLI

Para descrever uma autoridade de certificação privada

O `describe-certificate-authority` comando a seguir lista informações sobre a CA privada identificada peloARN.

```
aws acm-pca describe-certificate-authority --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012
```

- Para API obter detalhes, consulte [DescribeCertificateAuthority](#) na Referência de AWS CLI Comandos.

get-certificate-authority-certificate

O código de exemplo a seguir mostra como usar `get-certificate-authority-certificate`.

AWS CLI

Para recuperar um certificado de autoridade de certificação (CA)

O `get-certificate-authority-certificate` comando a seguir recupera o certificado e a cadeia de certificados da CA privada especificada peloARN.

```
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
```

- Para API obter detalhes, consulte [GetCertificateAuthorityCertificate](#) na Referência de AWS CLI Comandos.

get-certificate-authority-csr

O código de exemplo a seguir mostra como usar `get-certificate-authority-csr`.

AWS CLI

Para recuperar a solicitação de assinatura de certificado para uma autoridade de certificação

O `get-certificate-authority-csr` comando a seguir recupera o CSR para a CA privada especificada pelo 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
```

- Para API obter detalhes, consulte [GetCertificateAuthorityCsr](#) na Referência de AWS CLI Comandos.

get-certificate

O código de exemplo a seguir mostra como usar `get-certificate`.

AWS CLI

Para recuperar um certificado emitido

O `get-certificate` exemplo a seguir recupera um certificado da CA privada especificada.

```
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 text
```

Saída:

```
-----BEGIN CERTIFICATE-----
MIIEDzCCAvEgAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvzrZMwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMCVVMxEzARBgNVBAGMCl dhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqsKtu/KSS7YDjK0
KQNY73D6Ltmd0EbAyq10XiDxqY41lvKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE----- -----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIRA0skdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMCVVMxEzARBgNVBAGMCl dhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
kdRGB6P2hpxstDOUIwAoCbhoaWwFA4ybJznf+j0QhAziNlRdKQRR8nODWpKt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----
```

A primeira parte da saída é o próprio certificado. A segunda parte é a cadeia de certificados vinculada ao certificado CA raiz. Observe que, ao usar a `--output text` opção, um TAB caractere é inserido entre as duas partes do certificado (essa é a causa do texto recuado). Se você pretende pegar essa saída e analisar os certificados com outras ferramentas, talvez seja necessário remover o TAB caractere para que ele seja processado corretamente.

- Para API obter detalhes, consulte [GetCertificate](#) na Referência de AWS CLI Comandos.

import-certificate-authority-certificate

O código de exemplo a seguir mostra como usar `import-certificate-authority-certificate`.

AWS CLI

Para importar seu certificado de autoridade de certificação para ACM PCA

O `import-certificate-authority-certificate` comando a seguir importa o certificado de CA privado assinado para a CA especificada pelo 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
```

- Para API obter detalhes, consulte [ImportCertificateAuthorityCertificate](#) na Referência de AWS CLI Comandos.

issue-certificate

O código de exemplo a seguir mostra como usar `issue-certificate`.

AWS CLI

Para emitir um certificado privado

O `issue-certificate` comando a seguir usa a CA privada especificada pelo ARN para emitir um certificado privado.

```
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
```

- Para API obter detalhes, consulte [IssueCertificate](#) na Referência de AWS CLI Comandos.

list-certificate-authorities

O código de exemplo a seguir mostra como usar `list-certificate-authorities`.

AWS CLI

Para listar suas autoridades de certificação privadas

O `list-certificate-authorities` comando a seguir lista informações sobre todas as informações privadas CAs da sua conta.

```
aws acm-pca list-certificate-authorities --max-results 10
```

- Para API obter detalhes, consulte [ListCertificateAuthorities](#) na Referência de AWS CLI Comandos.

list-tags

O código de exemplo a seguir mostra como usar `list-tags`.

AWS CLI

Para listar as etiquetas da sua autoridade de certificação

O `list-tags` comando a seguir lista as tags associadas à CA privada especificada peloARN.

```
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
```

- Para API obter detalhes, consulte [ListTags](#) na Referência de AWS CLI Comandos.

revoke-certificate

O código de exemplo a seguir mostra como usar `revoke-certificate`.

AWS CLI

Para revogar um certificado privado

O `revoke-certificate` comando a seguir revoga um certificado privado da CA identificado peloARN.

```
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"
```

- Para API obter detalhes, consulte [RevokeCertificate](#) na Referência de AWS CLI Comandos.

tag-certificate-authority

O código de exemplo a seguir mostra como usar `tag-certificate-authority`.

AWS CLI

Para anexar etiquetas a uma autoridade de certificação privada

O `tag-certificate-authority` comando a seguir anexa uma ou mais tags à sua CA privada.

```
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
```

- Para API obter detalhes, consulte [TagCertificateAuthority](#) na Referência de AWS CLI Comandos.

untag-certificate-authority

O código de exemplo a seguir mostra como usar `untag-certificate-authority`.

AWS CLI

Para remover uma ou mais tags da sua autoridade de certificação privada

O `untag-certificate-authority` comando a seguir remove as tags da CA privada identificada pelo 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
```

- Para API obter detalhes, consulte [UntagCertificateAuthority](#) na Referência de AWS CLI Comandos.

update-certificate-authority

O código de exemplo a seguir mostra como usar `update-certificate-authority`.

AWS CLI

Para atualizar a configuração da sua autoridade de certificação privada

O `update-certificate-authority` comando a seguir atualiza o status e a configuração da CA privada identificada pelo 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"
```

- Para API obter detalhes, consulte [UpdateCertificateAuthority](#) na Referência de AWS CLI Comandos.

AWS Proton exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Proton.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-service-instance-deployment

O código de exemplo a seguir mostra como usar `cancel-service-instance-deployment`.

AWS CLI

Para cancelar a implantação de uma instância de serviço

O `cancel-service-instance-deployment` exemplo a seguir cancela a implantação de uma instância de serviço.

```
aws proton cancel-service-instance-deployment \  
  --service-instance-name "instance-one" \  
  --service-name "simple-svc"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Atualizar uma instância de serviço](#) no Guia do Administrador do AWS Proton ou [Atualizar uma instância de serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [CancelServiceInstanceDeployment](#) na Referência de AWS CLI Comandos.

cancel-service-pipeline-deployment

O código de exemplo a seguir mostra como usar `cancel-service-pipeline-deployment`.

AWS CLI

Para cancelar a implantação de um pipeline de serviços

O `cancel-service-pipeline-deployment` exemplo a seguir cancela a implantação de um pipeline de serviço.

```
aws proton cancel-service-pipeline-deployment \  
  --service-name "simple-svc"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Atualizar um pipeline de serviço](#) no Guia do Administrador do AWS Proton ou [Atualizar um pipeline de serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [CancelServicePipelineDeployment](#) na Referência de AWS CLI Comandos.

create-service

O código de exemplo a seguir mostra como usar create-service.

AWS CLI

Para criar um serviço

O create-service exemplo a seguir cria um serviço com um pipeline de serviços.

```
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
```

Conteúdo de 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"
```

Saída:

```
{  
  "service": {  
    "arn": "arn:aws:proton:region-id:123456789012:service/MySimpleService",  
    "createdAt": "2020-11-18T19:50:27.460000+00:00",  
    "lastModifiedAt": "2020-11-18T19:50:27.460000+00:00",  
    "name": "MySimpleService",  
    "repositoryConnectionArn": "arn:aws:codestar-connections:region-  
id:123456789012connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "repositoryId": "myorg/myapp",  
    "status": "CREATE_IN_PROGRESS",  
    "templateName": "fargate-service"  
  }  
}
```

Para obter mais informações, consulte [Criar um serviço](#) no Guia do Administrador do AWS Proton e [Criar um serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [CreateService](#) na Referência de AWS CLI Comandos.

delete-service

O código de exemplo a seguir mostra como usar delete-service.

AWS CLI

Como excluir um serviço

O `delete-service` exemplo a seguir exclui um serviço.

```
aws proton delete-service \  
  --name "simple-svc"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Excluir um serviço](#) no Guia do Administrador do AWS Proton.

- Para API obter detalhes, consulte [DeleteService](#) na Referência de AWS CLI Comandos.

get-service-instance

O código de exemplo a seguir mostra como usar `get-service-instance`.

AWS CLI

Para obter detalhes da instância de serviço

O `get-service-instance` exemplo a seguir obtém dados detalhados de uma instância de serviço.

```
aws proton get-service-instance \
  --name "instance-one" \
  --service-name "simple-svc"
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Exibir dados do serviço](#) no Guia do Administrador do AWS Proton ou [Exibir dados do serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [GetServiceInstance](#) na Referência de AWS CLI Comandos.

get-service

O código de exemplo a seguir mostra como usar `get-service`.

AWS CLI

Para obter detalhes do serviço

O `get-service` exemplo a seguir obtém dados detalhados de um serviço.

```
aws proton get-service \
  --name "simple-svc"
```

Saída:

```
{
  "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"
  }
}
```


Para obter mais informações, consulte [Exibir dados do serviço](#) no Guia do Administrador do AWS Proton ou [Exibir dados do serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [GetService](#) na Referência de AWS CLI Comandos.

list-service-instances

O código de exemplo a seguir mostra como usar `list-service-instances`.

AWS CLI

Exemplo 1: Para listar todas as instâncias de serviço

O `list-service-instances` exemplo a seguir lista as instâncias de serviço.

```
aws proton list-service-instances
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Exibir dados da instância de serviço](#) no Guia do Administrador do AWS Proton ou [Exibir dados da instância de serviço](#) no Guia do Usuário do AWS Proton.

Exemplo 2: Para listar a instância de serviço especificada

O `get-service-instance` exemplo a seguir obtém uma instância de serviço.

```
aws proton get-service-instance \  
  --name "instance-one" \  
  --service-name "simple-svc"
```

Saída:

```
{  
  "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:\nmy_sample_pipeline_optional_input: hello world\nmy_sample_pipeline_required_input: pipeline up\ninstances:\n- name: instance-one\nenvironment: 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"  
  }  
}
```

Para obter mais informações, consulte [Exibir dados da instância de serviço](#) no Guia do Administrador do AWS Proton ou [Exibir dados da instância de serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [ListServiceInstances](#) na Referência de AWS CLI Comandos.

update-service-instance

O código de exemplo a seguir mostra como usar `update-service-instance`.

AWS CLI

Para atualizar uma instância de serviço para uma nova versão secundária

O `update-service-instance` exemplo a seguir atualiza uma instância de serviço para uma nova versão secundária de seu modelo de serviço que adiciona uma nova instância chamada `my-other-instance` com uma nova entrada obrigatória.

```
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"
```

Conteúdo de `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"
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Atualizar uma instância de serviço](#) no Guia do Administrador do AWS Proton ou [Atualizar uma instância de serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [UpdateServiceInstance](#) na Referência de AWS CLI Comandos.

update-service-pipeline

O código de exemplo a seguir mostra como usar update-service-pipeline.

AWS CLI

Para atualizar um pipeline de serviços

O update-service-pipeline exemplo a seguir atualiza um pipeline de serviço para uma nova versão secundária de seu modelo de serviço.

```

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"

```

Saída:

```

{
  "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   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": "0",
    "templateName": "svc-simple"
  }
}

```

Para obter mais informações, consulte [Atualizar um pipeline de serviço](#) no Guia do Administrador do AWS Proton ou [Atualizar um pipeline de serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [UpdateServicePipeline](#) na Referência de AWS CLI Comandos.

update-service

O código de exemplo a seguir mostra como usar update-service.

AWS CLI

Para atualizar um serviço

O update-service exemplo a seguir edita a descrição de um serviço.

```

aws proton update-service \
  --name "MySimpleService" \
  --description "Edit by updating description"

```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Editar um serviço](#) no Guia do Administrador do AWS Proton ou [Editar um serviço](#) no Guia do Usuário do AWS Proton.

- Para API obter detalhes, consulte [UpdateService](#) na Referência de AWS CLI Comandos.

QLDBexemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface withQLDB.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-journal-kinesis-stream

O código de exemplo a seguir mostra como usar `cancel-journal-kinesis-stream`.

AWS CLI

Para cancelar uma transmissão do diário

O `cancel-journal-kinesis-stream` exemplo a seguir cancela o fluxo de diário especificado de um livro contábil.

```
aws qldb cancel-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISCKqwe4y25YyHLzYUFAf
```

Saída:

```
{  
  "StreamId": "7ISCKqwe4y25YyHLzYUFAf"  
}
```

Para obter mais informações, consulte [Dados de periódicos de streaming da Amazon QLDB](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [CancelJournalKinesisStream](#) na Referência de AWS CLI Comandos.

create-ledger

O código de exemplo a seguir mostra como usar `create-ledger`.

AWS CLI

Exemplo 1: como criar um livro contábil com propriedades padrão

O exemplo de `create-ledger` a seguir cria um livro contábil com o nome `myExampleLedger` e o modo de permissões `STANDARD`. Os parâmetros opcionais para proteção de exclusão e AWS KMS chave não são especificados, então eles usam como padrão uma chave própria `true` e uma KMS chave AWS própria, respectivamente.

```
aws qldb create-ledger \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

Saída:

```
{
  "State": "CREATING",
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "DeletionProtection": true,
  "CreationDateTime": 1568839243.951,
  "Name": "myExampleLedger",
  "PermissionsMode": "STANDARD"
}
```

Exemplo 2: Para criar um livro contábil com a proteção contra exclusão desativada, uma KMS chave gerenciada pelo cliente e tags especificadas

O exemplo de `create-ledger` a seguir cria um livro contábil com o nome `myExampleLedger2` e o modo de permissões `STANDARD`. O recurso de proteção contra exclusão está desativado, a KMS chave gerenciada pelo cliente especificada é usada para criptografia em repouso e as tags especificadas são anexadas ao recurso.

```
aws qldb create-ledger \
  --name myExampleLedger2 \
  --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
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Operações básicas do Amazon QLDB Ledgers](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [CreateLedger](#) na Referência de AWS CLI Comandos.

delete-ledger

O código de exemplo a seguir mostra como usar `delete-ledger`.

AWS CLI

Para excluir um livro contábil

O `delete-ledger` exemplo a seguir exclui o livro contábil especificado.

```
aws qlldb delete-ledger \  
  --name myExampleLedger
```

Este comando não produz saída.

Para obter mais informações, consulte [Operações básicas do Amazon QLDB Ledgers](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [DeleteLedger](#) na Referência de AWS CLI Comandos.

describe-journal-kinesis-stream

O código de exemplo a seguir mostra como usar `describe-journal-kinesis-stream`.

AWS CLI

Para descrever um fluxo de diário

O `describe-journal-kinesis-stream` exemplo a seguir exibe os detalhes do fluxo de diário especificado a partir de um livro contábil.

```
aws qlldb describe-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISCKqwe4y25YyHLzYUFAf
```

Saída:

```
{  
  "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: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"
  }
}

```

Para obter mais informações, consulte [Dados de periódicos de streaming da Amazon QLDB](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [DescribeJournalKinesisStream](#) na Referência de AWS CLI Comandos.

describe-journal-s3-export

O código de exemplo a seguir mostra como usar `describe-journal-s3-export`.

AWS CLI

Para descrever um trabalho de exportação de diário

O `describe-journal-s3-export` exemplo a seguir exibe os detalhes do trabalho de exportação especificado em um livro contábil.

```

aws qldb describe-journal-s3-export \
  --name myExampleLedger \
  --export-id ADR2ONPKN5LINYGb4dp7yZ

```

Saída:

```

{
  "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"
}
}
```

Para obter mais informações, consulte [Exportando seu periódico na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [DescribeJournalS3Export na Referência](#) de AWS CLI Comandos.

describe-ledger

O código de exemplo a seguir mostra como usar `describe-ledger`.

AWS CLI

Para descrever um livro contábil

O `describe-ledger` exemplo a seguir exibe os detalhes do livro contábil especificado.

```
aws qlldb describe-ledger \
  --name myExampleLedger
```

Saída:

```
{
  "CreationDateTime": 1568839243.951,
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",
  "State": "ACTIVE",
  "Name": "myExampleLedger",
```

```

    "DeletionProtection": true,
    "PermissionsMode": "STANDARD",
    "EncryptionDescription": {
      "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
      "EncryptionStatus": "ENABLED"
    }
  }
}

```

Para obter mais informações, consulte [Operações básicas do Amazon QLDB Ledgers](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [DescribeLedger](#) na Referência de AWS CLI Comandos.

export-journal-to-s3

O código de exemplo a seguir mostra como usar `export-journal-to-s3`.

AWS CLI

Para exportar blocos de diário para o S3

O `export-journal-to-s3` exemplo a seguir cria uma tarefa de exportação para blocos de diário dentro de um intervalo de data e hora especificado a partir de um livro contábil com o nome `myExampleLedger`. O trabalho de exportação grava os blocos em um bucket específico do Amazon S3.

```

aws qlldb export-journal-to-s3 \
  --name myExampleLedger \
  --inclusive-start-time 2019-09-18T00:00:00Z \
  --exclusive-end-time 2019-09-18T22:59:59Z \
  --role-arn arn:aws:iam::123456789012:role/my-s3-export-role \
  --s3-export-configuration file://my-s3-export-config.json

```

Conteúdo de `my-s3-export-config.json`:

```

{
  "Bucket": "awsExampleBucket",
  "Prefix": "ledgerexport1/",
  "EncryptionConfiguration": {
    "ObjectEncryptionType": "SSE_S3"
  }
}

```

```
}

```

Saída:

```
{
  "ExportId": "ADR20NPKN5LINYGb4dp7yZ"
}
```

Para obter mais informações, consulte [Exportando seu periódico na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [ExportJournalToS3](#) na Referência de AWS CLI Comandos.

get-block

O código de exemplo a seguir mostra como usar `get-block`.

AWS CLI

Exemplo 1: Para obter um bloco de diário e uma prova para verificação usando arquivos de entrada

O `get-block` exemplo a seguir solicita um objeto de dados de bloco e uma prova do livro contábil especificado. A solicitação é para um endereço de dica de resumo e endereço de bloco especificados.

```
aws qlldb get-block \
  --name vehicle-registration \
  --block-address file://myblockaddress.json \
  --digest-tip-address file://mydigesttipaddress.json
```

Conteúdo de `myblockaddress.json`:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"
}
```

Conteúdo de `mydigesttipaddress.json`:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"
}
```

}

Saída:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iwl
\\",sequenceNo:100},transactionId:\\"FnQeJBAicTX0Ah32ZnVtSX
\\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{105L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdBc1cZKxhVmpoMHPUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDD5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yJHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},
{{y5cCB7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}}],transactionInfo:{statements:[{statement:
\\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\\
\\nINSERT INTO r.Owners.SecondaryOwners\\n  VALUE { 'PersonId' :
'CMVdR77XP8zAg1mmFDGTvt' }\\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vs0J0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\\"VehicleRegistration\\",tableId:
\\"BFJKdXgzT9oF4wjMbuXy4G\\",statements:[0]}},revisions:[{blockAddress:
{strandId:\\"KmA3ZZca7vAIiJAK9S5Iwl\\",sequenceNo:100},hash:
{{mHVex/yJHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},data:{VIN:
\\"1N4AL11D75C109151\\",LicensePlateNumber:\\"LEWISR261LL\\",State:\\"WA
\\",PendingPenaltyTicketAmount:90.25,ValidFromDate:2017-08-21,ValidToDate:2020-05-11,Owners:
{PrimaryOwner:{PersonId:\\"BFJKdXhnLRT27sXBnojNGW\\"},SecondaryOwners:
[{{PersonId:\\"CMVdR77XP8zAg1mmFDGTvt\\"}]}],City:\\"Everett\\"},metadata:{id:
\\"JUJgkIcNbhS2goq8RqLuZ4\\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\\"FnQeJBAicTX0Ah32ZnVtSX\\"}}}]"}
  },
  "Proof": {
    "IonText": "[{{13+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhz1nGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
+qE=}}]"
  }
}
```

Para obter mais informações, consulte [Verificação de dados na Amazon QLDB no Amazon QLDB Developer Guide](#).

Exemplo 2: Para obter um bloco de diário e uma prova para verificação usando a sintaxe abreviada

O `get-block` exemplo a seguir solicita um objeto de dados de bloco e uma prova do livro contábil especificado usando a sintaxe abreviada. A solicitação é para um endereço de dica de resumo e endereço de bloco especificados.

```
aws qlldb get-block \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"'
  \
  --digest-tip-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"'
```

Saída:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100},transactionId:\\"FnQeJBAicTX0Ah32ZnVtSX\\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:{{NoChM92yKRuJAb/jeLd1VnYn4DHWiF071ACfic9uHc=}},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'\\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/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=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkN0xoI=}}, {{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}}, {{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
+qE=}}]"
}
}

```

Para obter mais informações, consulte [Verificação de dados na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [GetBlock](#) na Referência de AWS CLI Comandos.

get-digest

O código de exemplo a seguir mostra como usar `get-digest`.

AWS CLI

Para obter um resumo de um livro contábil

O `get-digest` exemplo a seguir solicita um resumo do livro contábil especificado no último bloco confirmado no diário.

```

aws qlldb get-digest \
  --name vehicle-registration

```

Saída:

```

{
  "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",
  "DigestTipAddress": {
    "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\", sequenceNo:123}"
  }
}

```

Para obter mais informações, consulte [Verificação de dados na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [GetDigest](#) na Referência de AWS CLI Comandos.

get-revision

O código de exemplo a seguir mostra como usar `get-revision`.

AWS CLI

Exemplo 1: Para obter uma revisão do documento e uma prova para verificação usando arquivos de entrada

O `get-revision` exemplo a seguir solicita um objeto de dados de revisão e uma prova do livro contábil especificado. A solicitação é para um endereço de dica de resumo, ID do documento e endereço de bloco especificados da revisão.

```
aws qldb get-revision \
  --name vehicle-registration \
  --block-address file://myblockaddress.json \
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \
  --digest-tip-address file://mydigesttipaddress.json
```

Conteúdo de `myblockaddress.json`:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iwl\\",sequenceNo:100}"
}
```

Conteúdo de `mydigesttipaddress.json`:

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iwl\\",sequenceNo:123}"
}
```

Saída:

```
{
  "Revision": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5Iwl\\",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": "[{{eRSwnmAM7WWANWdD5iG0yK+T4tDXyzUq6HZ/0fgLHos=}}, {{VV1rdaNuf
+yJZVGlmsM6gr2T52QvB08Lg+KgpjcnWAU=}},
{{7kewBXhpdBc1cZKxhVmpoMhpUG0JtwQD0iY2LPfZkYA=}}, {{13+EXs69K1+reh1qyWlkt
+oHD1w4Zi9pCLW/t/mgTPM=}}, {{48CXG3ehPqsxCYd34EEa8Fso00RpWAA08010RJkF3Do=}},
{{9UnwnKSQT0i3ge1JMva+tMIqCEDaOPTkwxmyHSn8UPQ=}}, {{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNECb1sCwcDaI=}}, {{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkN0xoI=}}, {{+3CwpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfy=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
  }
}
```

Para obter mais informações, consulte [Verificação de dados na Amazon QLDB no Amazon QLDB Developer Guide](#).

Exemplo 2: Para obter uma revisão do documento e uma prova para verificação usando a sintaxe abreviada

O `get-revision` exemplo a seguir solicita um objeto de dados de revisão e uma prova do livro contábil especificado usando a sintaxe abreviada. A solicitação é para um endereço de dica de resumo, ID do documento e endereço de bloco especificados da revisão.

```
aws qldb get-revision \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:\ "KmA3ZZca7vAIiJAK9S5Iw1", sequenceNo:100}"'
  \
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \
  --digest-tip-address 'IonText="{strandId:\ "KmA3ZZca7vAIiJAK9S5Iw1
  \", sequenceNo:123}"'
```

Saída:

```
{
  "Revision": {
    "IonText": "{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":"CMVdR77XP8zAglmmFDGTvt\"}],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=}}]\"
  }
}
```

Para obter mais informações, consulte [Verificação de dados na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [GetRevision](#) na Referência de AWS CLI Comandos.

list-journal-kinesis-streams-for-ledger

O código de exemplo a seguir mostra como usar `list-journal-kinesis-streams-for-ledger`.

AWS CLI

Para listar fluxos de diário para um livro contábil

O `list-journal-kinesis-streams-for-ledger` exemplo a seguir lista fluxos de diário para o livro contábil especificado.

```
aws qldb list-journal-kinesis-streams-for-ledger \
  --ledger-name myExampleLedger
```

Saída:

```
{
  \"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"
  }
]
}

```

Para obter mais informações, consulte [Dados de periódicos de streaming da Amazon QLDB](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [ListJournalKinesisStreamsForLedger](#) na Referência de AWS CLI Comandos.

list-journal-s3-exports-for-ledger

O código de exemplo a seguir mostra como usar `list-journal-s3-exports-for-ledger`.

AWS CLI

Para listar trabalhos de exportação de diário para um livro contábil

O `list-journal-s3-exports-for-ledger` exemplo a seguir lista os trabalhos de exportação de diário para o livro contábil especificado.

```

aws qldb list-journal-s3-exports-for-ledger \
  --name myExampleLedger

```

Saída:

```

{
  "JournalS3Exports": [

```

```
{
  "LedgerName": "myExampleLedger",
  "ExclusiveEndTime": 1568847599.0,
  "ExportCreationTime": 1568847801.418,
  "S3ExportConfiguration": {
    "Bucket": "awsExampleBucket",
    "Prefix": "ledgerexport1/",
    "EncryptionConfiguration": {
      "ObjectEncryptionType": "SSE_S3"
    }
  },
  "ExportId": "ADR20NPKN5LINYGb4dp7yZ",
  "RoleArn": "arn:aws:iam::123456789012:role/qlldb-s3-export",
  "InclusiveStartTime": 1568764800.0,
  "Status": "IN_PROGRESS"
}
]
```

Para obter mais informações, consulte [Exportando seu periódico na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [ListJournalS3 ExportsForLedger](#) na Referência de AWS CLI Comandos.

list-journal-s3-exports

O código de exemplo a seguir mostra como usar `list-journal-s3-exports`.

AWS CLI

Para listar trabalhos de exportação de diário

O `list-journal-s3-exports` exemplo a seguir lista os trabalhos de exportação de diário para todos os livros contábeis associados à AWS conta corrente e à região.

```
aws qlldb list-journal-s3-exports
```

Saída:

```
{
  "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
    }
  ]
}

```

Para obter mais informações, consulte [Exportando seu periódico na Amazon QLDB no Amazon QLDB Developer Guide](#).

- Para API obter detalhes, consulte [ListJournalS3Exports na Referência](#) de AWS CLI Comandos.

list-ledgers

O código de exemplo a seguir mostra como usar `list-ledgers`.

AWS CLI

Como listar os livros de registros disponíveis

O `list-ledgers` exemplo a seguir lista todos os livros contábeis associados à AWS conta corrente e à região.

```
aws qlldb list-ledgers
```

Saída:

```
{
  "Ledgers": [
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839243.951,
      "Name": "myExampleLedger"
    },
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839543.557,
      "Name": "myExampleLedger2"
    }
  ]
}
```

Para obter mais informações, consulte [Operações básicas do Amazon QLDB Ledgers](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [ListLedgers](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as etiquetas anexadas a um livro contábil

O `list-tags-for-resource` exemplo a seguir lista todas as tags anexadas ao livro contábil especificado.

```
aws qlldb list-tags-for-resource \  
  --resource-arn arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger
```

Saída:

```
{  
  "Tags": {  
    "IsTest": "true",  
    "Domain": "Test"  
  }  
}
```

Para obter mais informações, consulte Como [marcar QLDB recursos da Amazon](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

stream-journal-to-kinesis

O código de exemplo a seguir mostra como usar `stream-journal-to-kinesis`.

AWS CLI

Exemplo 1: Para transmitir dados do diário para o Kinesis Data Streams usando arquivos de entrada

O `stream-journal-to-kinesis` exemplo a seguir cria um fluxo de dados do diário dentro de um intervalo de data e hora especificado a partir de um livro contábil com o nome `myExampleLedger`. O stream envia os dados para um stream de dados específico do Amazon Kinesis.

```
aws qlldb stream-journal-to-kinesis \  
  --ledger-name myExampleLedger \  
  --inclusive-start-time 2020-05-29T00:00:00Z \  
  --exclusive-end-time 2020-05-29T23:59:59Z \  
  --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \  
  --kinesis-configuration file://my-kinesis-config.json \  
  --stream-name myExampleLedger-stream
```

Conteúdo de `my-kinesis-config.json`:


```
{
  "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",
  "AggregationEnabled": true
}
```

Saída:

```
{
  "StreamId": "7ISckqwe4y25YyHLzYUFaf"
}
```

Para obter mais informações, consulte [Dados de periódicos de streaming da Amazon QLDB](#) no Amazon QLDB Developer Guide.

Exemplo 2: Para transmitir dados do diário para o Kinesis Data Streams usando a sintaxe abreviada

O `stream-journal-to-kinesis` exemplo a seguir cria um fluxo de dados do diário dentro de um intervalo de data e hora especificado a partir de um livro contábil com o nome `myExampleLedger`. O stream envia os dados para um stream de dados específico do Amazon Kinesis.

```
aws qlldb stream-journal-to-kinesis \
  --ledger-name myExampleLedger \
  --inclusive-start-time 2020-05-29T00:00:00Z \
  --exclusive-end-time 2020-05-29T23:59:59Z \
  --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
  --stream-name myExampleLedger-stream \
  --kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb,AggregationEnabled=true
```

Saída:

```
{
  "StreamId": "7ISckqwe4y25YyHLzYUFaf"
}
```

Para obter mais informações, consulte [Dados de periódicos de streaming da Amazon QLDB](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [StreamJournalToKinesis](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar um livro contábil

O `tag-resource` exemplo a seguir adiciona um conjunto de tags a um livro contábil especificado.

```
aws qldb tag-resource \  
  --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tags IsTest=true,Domain=Test
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar QLDB recursos da Amazon](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Como remover as tags de um recurso

O `untag-resource` exemplo a seguir remove tags com as chaves de tag especificadas de um livro contábil especificado.

```
aws qldb untag-resource \  
  --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tag-keys IsTest Domain
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar QLDB recursos da Amazon](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-ledger-permissions-mode

O código de exemplo a seguir mostra como usar `update-ledger-permissions-mode`.

AWS CLI

Exemplo 1: Para atualizar o modo de permissões de um livro contábil para STANDARD

O `update-ledger-permissions-mode` exemplo a seguir atribui o modo de STANDARD permissões ao livro contábil especificado.

```
aws qlldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

Saída:

```
{  
  "Name": "myExampleLedger",  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "PermissionsMode": "STANDARD"  
}
```

Exemplo 2: Para atualizar o modo de permissões de um livro contábil para `_ ALLOW ALL`

O `update-ledger-permissions-mode` exemplo a seguir atribui o modo de `ALLOW_ALL` permissões ao livro contábil especificado.

```
aws qlldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode ALLOW_ALL
```

Saída:

```
{
```

```
"Name": "myExampleLedger",
"Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
"PermissionsMode": "ALLOW_ALL"
}
```

Para obter mais informações, consulte [Operações básicas do Amazon QLDB Ledgers](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [UpdateLedgerPermissionsModena](#) Referência de AWS CLI Comandos.

update-ledger

O código de exemplo a seguir mostra como usar `update-ledger`.

AWS CLI

Exemplo 1: Para atualizar a propriedade de proteção contra exclusão de um livro contábil

O `update-ledger` exemplo a seguir atualiza o livro contábil especificado para desativar o recurso de proteção contra exclusão.

```
aws qldb update-ledger \
  --name myExampleLedger \
  --no-deletion-protection
```

Saída:

```
{
  "CreationDateTime": 1568839243.951,
  "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
  "DeletionProtection": false,
  "Name": "myExampleLedger",
  "State": "ACTIVE"
}
```

Exemplo 2: Para atualizar a AWS KMS chave de um livro contábil para uma chave gerenciada pelo cliente

O `update-ledger` exemplo a seguir atualiza o livro contábil especificado para usar uma KMS chave gerenciada pelo cliente para criptografia em repouso.

```
aws qlldb update-ledger \  
  --name myExampleLedger \  
  --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "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"  
  }  
}
```

Exemplo 3: Para atualizar a AWS KMS chave de um livro contábil para uma chave AWS própria

O `update-ledger` exemplo a seguir atualiza o livro contábil especificado para usar uma AWS KMS chave própria para criptografia em repouso.

```
aws qlldb update-ledger \  
  --name myExampleLedger \  
  --kms-key AWS_OWNED_KMS_KEY
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Operações básicas do Amazon QLDB Ledgers](#) no Amazon QLDB Developer Guide.

- Para API obter detalhes, consulte [UpdateLedger](#) na Referência de AWS CLI Comandos.

RDSExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonRDS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-option-to-option-group

O código de exemplo a seguir mostra como usar `add-option-to-option-group`.

AWS CLI

Para adicionar uma opção a um grupo de opções

O `add-option-to-option-group` exemplo a seguir adiciona uma opção ao grupo de opções especificado.

```
aws rds add-option-to-option-group \  
  --option-group-name myoptiongroup \  
  --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \  
  --apply-immediately
```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Adicionar uma opção a um grupo de opções](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [AddOptionToOptionGroup](#) na Referência de AWS CLI Comandos.

add-role-to-db-cluster

O código de exemplo a seguir mostra como usar `add-role-to-db-cluster`.

AWS CLI

Para associar uma função de AWS Identity and Access Management (IAM) a um cluster de banco de dados

O `add-role-to-db-cluster` exemplo a seguir associa uma função a um cluster de banco de dados.

```
aws rds add-role-to-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Este comando não produz saída.

Para obter mais informações, consulte [Associar uma IAM função a um cluster Amazon Aurora SQL My DB](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [AddRoleToDbCluster](#) na Referência de AWS CLI Comandos.

`add-role-to-db-instance`

O código de exemplo a seguir mostra como usar `add-role-to-db-instance`.

AWS CLI

Para associar uma função de AWS Identity and Access Management (IAM) a uma instância de banco de dados

O `add-role-to-db-instance` exemplo a seguir adiciona a função a uma instância de banco de dados Oracle chamada `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Pré-requisitos para a integração do Amazon RDS Oracle com o Amazon S3 no Guia do usuário da Amazon](#). RDS

- Para API obter detalhes, consulte [AddRoleToDbInstance](#) na Referência de AWS CLI Comandos.

add-source-identifier-to-subscription

O código de exemplo a seguir mostra como usar `add-source-identifier-to-subscription`.

AWS CLI

Para adicionar um identificador de origem a uma assinatura

O `add-source-identifier` exemplo a seguir adiciona outro identificador de origem a uma assinatura existente.

```
aws rds add-source-identifier-to-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

- Para API obter detalhes, consulte [AddSourceIdentifierToSubscription](#) na Referência de AWS CLI Comandos.

add-tags-to-resource

O código de exemplo a seguir mostra como usar `add-tags-to-resource`.

AWS CLI

Para adicionar tags a um recurso

O `add-tags-to-resource` exemplo a seguir adiciona tags a um RDS banco de dados.

```
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\"}]"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar RDS recursos da Amazon](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [AddTagsToResource](#) na Referência de AWS CLI Comandos.

apply-pending-maintenance-action

O código de exemplo a seguir mostra como usar `apply-pending-maintenance-action`.

AWS CLI

Para aplicar ações de manutenção pendentes

O `apply-pending-maintenance-action` exemplo a seguir aplica as ações de manutenção pendentes para um cluster de banco de dados.

```
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
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Manutenção de uma instância](#) de banco de dados no Guia RDS do usuário da Amazon e [Manutenção de um cluster de banco de dados Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ApplyPendingMaintenanceAction](#) na Referência de AWS CLI Comandos.

authorize-db-security-group-ingress

O código de exemplo a seguir mostra como usar `authorize-db-security-group-ingress`.

AWS CLI

Para associar uma função de AWS Identity and Access Management (IAM) a uma instância de banco de dados

O `authorize-db-security-group-ingress` exemplo a seguir configura o grupo de segurança padrão com uma regra de entrada para o intervalo de 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
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Autorizando o acesso à rede a um grupo de segurança de banco de dados a partir de um intervalo de IP](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [AuthorizeDbSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

backtrack-db-cluster

O código de exemplo a seguir mostra como usar `backtrack-db-cluster`.

AWS CLI

Para retroceder um cluster de banco de dados Aurora

O `backtrack-db-cluster` exemplo a seguir retrocede o cluster de banco de dados especificado para 19 de março de 2018, às 10h.

```
aws rds backtrack-db-cluster --db-cluster-identifier sample-cluster --backtrack-to 2018-03-19T10:00:00+00:00
```

Esse comando gera um JSON bloco que reconhece a alteração no recurso. RDS

- Para API obter detalhes, consulte [BacktrackDbCluster](#) na Referência de AWS CLI Comandos.

cancel-export-task

O código de exemplo a seguir mostra como usar `cancel-export-task`.

AWS CLI

Para cancelar a exportação de um snapshot para o Amazon S3

O `cancel-export-task` exemplo a seguir cancela uma tarefa de exportação em andamento que está exportando um snapshot para o Amazon S3.

```
aws rds cancel-export-task \  
  --export-task-identifier my-s3-export-1
```

Saída:

```
{  
  "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  
}
```

Para obter mais informações, consulte [Cancelamento de uma tarefa de exportação de instantâneos](#) no Guia RDS do usuário da Amazon ou [Cancelamento de uma tarefa de exportação de instantâneos](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CancelExportTask](#) na Referência de AWS CLI Comandos.

copy-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `copy-db-cluster-parameter-group`.

AWS CLI

Para copiar um grupo de parâmetros do cluster de banco de dados

O `copy-db-cluster-parameter-group` exemplo a seguir faz uma cópia de um grupo de parâmetros do cluster de banco de dados.

```
aws rds copy-db-cluster-parameter-group \  
  --source-db-cluster-parameter-group-identifier mydbclusterpg \  
  --target-db-cluster-parameter-group-identifier mydbclusterpgcopy \  
  --target-db-cluster-parameter-group-description "Copy of mydbclusterpg parameter group"
```

Saída:

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "mydbclusterpgcopy",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-pg:mydbclusterpgcopy",  
    "DBParameterGroupFamily": "aurora-mysql5.7",  
    "Description": "Copy of mydbclusterpg parameter group"  
  }  
}
```

Para obter mais informações, consulte [Copiar um grupo de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CopyDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

copy-db-cluster-snapshot

O código de exemplo a seguir mostra como usar `copy-db-cluster-snapshot`.

AWS CLI

Para copiar um snapshot de cluster de banco de dados

O `copy-db-cluster-snapshot` exemplo a seguir cria uma cópia de um snapshot de cluster de banco de dados, incluindo suas tags.

```
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 \  
  --target-db-cluster-snapshot-description "Copy of myclustersnapshotcopy"
```

--copy-tags

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Cópia de um snapshot](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CopyDbClusterSnapshot](#) na Referência de AWS CLI Comandos.

copy-db-parameter-group

O código de exemplo a seguir mostra como usar `copy-db-parameter-group`.

AWS CLI

Para copiar um grupo de parâmetros do cluster de banco de dados

O `copy-db-parameter-group` exemplo a seguir faz uma cópia de um grupo de parâmetros de banco de dados.

```
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"
```

Saída:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbpgcopy",  
    "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",  
    "DBParameterGroupFamily": "mysql5.7",  
    "Description": "Copy of mydbpg parameter group"  
  }  
}
```

Para obter mais informações, consulte [Copiar um grupo de parâmetros de banco](#) de dados no Amazon RDS Users Guide.

- Para API obter detalhes, consulte [CopyDbParameterGroup](#) na Referência de AWS CLI Comandos.

copy-db-snapshot

O código de exemplo a seguir mostra como usar `copy-db-snapshot`.

AWS CLI

Para copiar um DB snapshot

O `copy-db-snapshot` exemplo a seguir cria uma cópia de um DB snapshot.

```
aws rds copy-db-snapshot \  
  --source-db-snapshot-identifier rds:database-mysql-2019-06-06-08-38  
  --target-db-snapshot-identifier mydbsnapshotcopy
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Cópia de um snapshot](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [CopyDbSnapshot](#) na Referência de AWS CLI Comandos.

copy-option-group

O código de exemplo a seguir mostra como usar copy-option-group.

AWS CLI

Para copiar um grupo de opções

O `copy-option-group` exemplo a seguir faz uma cópia de um grupo de opções.

```
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"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte Como [fazer uma cópia de um grupo de opções](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [CopyOptionGroup](#) na Referência de AWS CLI Comandos.

create-blue-green-deployment

O código de exemplo a seguir mostra como usar `create-blue-green-deployment`.

AWS CLI

Exemplo 1: Para criar uma implantação azul/verde para uma instância RDS for My DB SQL

O `create-blue-green-deployment` exemplo a seguir cria uma implantação azul/verde para uma SQL instância My DB.

```
aws rds create-blue-green-deployment \  
  --source-option-group-identifier myoptiongroup \  
  --target-option-group-identifier new-option-group \  
  --target-option-group-description "My option group copy"
```

```
--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
```

Saída:

```
{  
  "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"
}
}

```

Para obter mais informações, consulte [Criação de uma implantação azul/verde no Guia RDS](#) do usuário da Amazon.

Exemplo 2: Para criar uma implantação azul/verde para um cluster Aurora My DB SQL

O create-blue-green-deployment exemplo a seguir cria uma implantação azul/verde para um 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

```

Saída:

```

{
  "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:mysql-
cluster-2",
      "Status": "PROVISIONING"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:db: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"
}
```

```
}
```

Para obter mais informações, consulte [Criação de uma implantação azul/verde no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateBlueGreenDeployment](#) na Referência de AWS CLI Comandos.

create-db-cluster-endpoint

O código de exemplo a seguir mostra como usar `create-db-cluster-endpoint`.

AWS CLI

Para criar um endpoint de cluster de banco de dados personalizado

O `create-db-cluster-endpoint` exemplo a seguir cria um endpoint de cluster de banco de dados personalizado e o associa ao cluster de banco de dados Aurora especificado.

```
aws rds create-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --endpoint-type reader \  
  --db-cluster-identifier mydbcluster \  
  --static-members dbinstance1 dbinstance2
```

Saída:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnpxexample.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"
```

```
}
```

Para obter mais informações, consulte [Amazon Aurora Connection Management](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateDbClusterEndpoint](#) na Referência de AWS CLI Comandos.

create-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `create-db-cluster-parameter-group`.

AWS CLI

Para criar um grupo de parâmetros de cluster de banco de dados

O `create-db-cluster-parameter-group` exemplo a seguir cria um grupo de parâmetros de cluster de banco de dados.

```
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"
```

Saída:

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "mydbclusterparametergroup",  
    "DBParameterGroupFamily": "aurora5.6",  
    "Description": "My new cluster parameter group",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
pg:mydbclusterparametergroup"  
  }  
}
```

Para obter mais informações, consulte [Criação de um grupo de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

create-db-cluster-snapshot

O código de exemplo a seguir mostra como usar `create-db-cluster-snapshot`.

AWS CLI

Para criar um snapshot de cluster de banco de dados

O `create-db-cluster-snapshot` exemplo a seguir cria um snapshot de cluster de banco de dados.

```
aws rds create-db-cluster-snapshot \  
  --db-cluster-identifier mydbcluster \  
  --db-cluster-snapshot-identifier mydbclustersnapshot
```

Saída:

```
{  
  "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
```

```
}  
}
```

Para obter mais informações, consulte [Criação de um snapshot de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateDbClusterSnapshot](#) na Referência de AWS CLI Comandos.

create-db-cluster

O código de exemplo a seguir mostra como usar `create-db-cluster`.

AWS CLI

Exemplo 1: Para criar um cluster de banco de dados compatível com My SQL 5.7

O `create-db-cluster` exemplo a seguir cria um cluster de banco de dados SQL compatível com My 5.7 usando a versão padrão do mecanismo. Substitua a senha `secret99` de amostra por uma senha segura. Quando você usa o console para criar um cluster de banco de dados, a Amazon cria RDS automaticamente a instância de banco de dados gravadora para seu cluster de banco de dados. No entanto, ao usar o AWS CLI para criar um cluster de banco de dados, você deve criar explicitamente a instância de banco de dados gravadora para seu cluster de banco de dados usando o `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
```

Saída:

```
{  
  "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-cnpxample.us-east-1.rds.amazonaws.com",
"DBClusterParameterGroup": "default.aurora-mysql5.7",
"HostedZoneId": "Z2R2ITUGPM61AM",
"ReaderEndpoint": "sample-cluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
"CopyTagsToSnapshot": false
}
}

```

Exemplo 2: Para criar um cluster de banco de dados SQL compatível com Postgre

O `create-db-cluster` exemplo a seguir cria um cluster de banco de dados SQL compatível com Postgre usando a versão padrão do mecanismo. Substitua a senha `secret99` de exemplo

por uma senha segura. Quando você usa o console para criar um cluster de banco de dados, a Amazon cria RDS automaticamente a instância de banco de dados gravadora para seu cluster de banco de dados. No entanto, ao usar o AWS CLI para criar um cluster de banco de dados, você deve criar explicitamente a instância de banco de dados gravadora para seu cluster de banco de dados usando o `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
```

Saída:

```
{  
  "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-cnpxexample.us-
east-1.rds.amazonaws.com",
    "MasterUsername": "master",
    "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c"
    ],
    "ReadReplicaIdentifiers": [],
    "DBSubnetGroup": "default"
}
}
```

Para obter mais informações, consulte [Criação de um cluster de banco de dados Amazon Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateDbCluster](#) na Referência de AWS CLI Comandos.

create-db-instance-read-replica

O código de exemplo a seguir mostra como usar `create-db-instance-read-replica`.

AWS CLI

Para criar uma instância de banco de dados, leia a réplica

Este exemplo cria uma réplica de leitura de uma instância de banco de dados existente chamada `test-instance`. A réplica de leitura tem um nome `test-instance-repl`.

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier test-instance-repl \
  --source-db-instance-identifier test-instance
```

Saída:

```
{
  "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...
  }
}
```

- Para API obter detalhes, consulte [CreateDbInstanceReadReplica](#) na Referência de AWS CLI Comandos.

create-db-instance

O código de exemplo a seguir mostra como usar `create-db-instance`.

AWS CLI

Como criar uma instância de banco de dados

O exemplo de `create-db-instance` a seguir usa as opções necessárias para iniciar uma nova instância de banco de dados.

```
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
```

Saída:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "test-mysql-instance",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "creating",
    "MasterUsername": "admin",
```

```
"AllocatedStorage": 20,
"PreferredBackupWindow": "12:55-13:25",
"BackupRetentionPeriod": 1,
"DBSecurityGroups": [],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-12345abc",
    "Status": "active"
  }
],
"DBParameterGroups": [
  {
    "DBParameterGroupName": "default.mysql5.7",
    "ParameterApplyStatus": "in-sync"
  }
],
"DBSubnetGroup": {
  "DBSubnetGroupName": "default",
  "DBSubnetGroupDescription": "default",
  "VpcId": "vpc-2ff2ff2f",
  "SubnetGroupStatus": "Complete",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2c"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2d"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-#####",
      "SubnetAvailabilityZone": {
        "Name": "us-west-2a"
      },
      "SubnetStatus": "Active"
    }
  ]
}
```

```

        "SubnetIdentifier": "subnet-#####",
        "SubnetAvailabilityZone": {
            "Name": "us-west-2b"
        },
        "SubnetStatus": "Active"
    }
]
},
"PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
"PendingModifiedValues": {
    "MasterUserPassword": "*****"
},
"MultiAZ": false,
"EngineVersion": "5.7.22",
"AutoMinorVersionUpgrade": true,
"ReadReplicaDBInstanceIdentifiers": [],
"LicenseModel": "general-public-license",
"OptionGroupMemberships": [
    {
        "OptionGroupName": "default:mysql-5-7",
        "Status": "in-sync"
    }
],
"PubliclyAccessible": true,
"StorageType": "gp2",
"DbInstancePort": 0,
"StorageEncrypted": false,
"DbiResourceId": "db-5555EXAMPLE444444444EXAMPLE",
"CACertificateIdentifier": "rds-ca-2019",
"DomainMemberships": [],
"CopyTagsToSnapshot": false,
"MonitoringInterval": 0,
"DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
"IAMDatabaseAuthenticationEnabled": false,
"PerformanceInsightsEnabled": false,
"DeletionProtection": false,
"AssociatedRoles": []
}
}

```

Para obter mais informações, consulte [Criação de uma RDS instância](#) de banco de dados da Amazon no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [C reateDBInstance](#) na Referência de AWS CLI Comandos.

create-db-parameter-group

O código de exemplo a seguir mostra como usar `create-db-parameter-group`.

AWS CLI

Como criar um grupo de parâmetros de banco de dados

O exemplo de `create-db-parameter-group` a seguir cria um grupo de parâmetros de banco de dados.

```
aws rds create-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup \  
  --db-parameter-group-family MySQL5.6 \  
  --description "My new parameter group"
```

Saída:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbparametergroup",  
    "DBParameterGroupFamily": "mysql5.6",  
    "Description": "My new parameter group",  
    "DBParameterGroupArn": "arn:aws:rds:us-  
east-1:123456789012:pg:mydbparametergroup"  
  }  
}
```

Para obter mais informações, consulte [Criação de um grupo de parâmetros de banco de dados](#) de dados no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [reateDBParameterGrupo C](#) na Referência de AWS CLI Comandos.

create-db-proxy-endpoint

O código de exemplo a seguir mostra como usar `create-db-proxy-endpoint`.

AWS CLI

Para criar um endpoint de proxy de banco de dados para um RDS banco de dados

O `create-db-proxy-endpoint` exemplo a seguir cria um endpoint de proxy de banco de dados.

```
aws rds create-db-proxy-endpoint \  
  --db-proxy-name proxyExample \  
  --db-proxy-endpoint-name "proxyep1" \  
  --vpc-subnet-ids subnetgroup1 subnetgroup2
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Criação de um endpoint proxy](#) no Guia do RDS usuário da Amazon e [Criação de um endpoint proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateDbProxyEndpoint](#) na Referência de AWS CLI Comandos.

create-db-proxy

O código de exemplo a seguir mostra como usar `create-db-proxy`.

AWS CLI

Para criar um proxy de banco de dados para um RDS banco de dados

O `create-db-proxy` exemplo a seguir cria um proxy de banco de dados.

```
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="MYSQ
\
  --role-arn arn:aws:iam::123456789123:role/ProxyRole \
  --vpc-subnet-ids subnetgroup1 subnetgroup2
```

Saída:

```
{
  "DBProxy": {
    "DBProxyName": "proxyExample",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
    "EngineFamily": "MYSQL",
    "VpcId": "vpc-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234",
      "sg-5678",
      "sg-9101"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
        "Description": "proxydescription1",
        "AuthScheme": "SECRETS",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
```

```

        "IAMAuth": "DISABLED"
      }
    ],
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyRole",
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "RequireTLS": false,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdatedAt": "2023-04-13T01:49:38.568000+00:00"
  }
}

```

Para obter mais informações, consulte [Criação de um RDS proxy](#) no Guia do RDS usuário da Amazon e [Criação de um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [CreateDbProxy](#) na Referência de AWS CLI Comandos.

create-db-security-group

O código de exemplo a seguir mostra como usar create-db-security-group.

AWS CLI

Para criar um grupo de segurança do Amazon RDS DB

O create-db-security-group comando a seguir cria um novo grupo de segurança do Amazon RDS DB:

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-group-description "My Test Security Group"
```

No exemplo, o novo grupo de segurança do banco de dados é nomeado mysecgroup e tem uma descrição.

Saída:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [CreateDbSecurityGroup](#) na Referência de AWS CLI Comandos.

create-db-shard-group

O código de exemplo a seguir mostra como usar `create-db-shard-group`.

AWS CLI

Exemplo 1: Para criar um cluster de banco de dados primário do Aurora Postgre SQL

O `create-db-cluster` exemplo a seguir cria um cluster de banco de dados SQL SQL primário do Aurora Postgre compatível com o Aurora Serverless v2 e o 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

```

Saída:

```

{
  "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-cekyexample.us-
east-2.rds.amazonaws.com",
"ReaderEndpoint": "my-sv2-cluster.cluster-ro-cekyexample.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
    }
}
}
}

```

Exemplo 2: Para criar a instância de banco de dados primária (gravadora)

O `create-db-instance` exemplo a seguir cria uma instância de banco de dados primária (gravadora) do Aurora Serverless v2. Quando você usa o console para criar um cluster de banco de dados, a Amazon cria RDS automaticamente a instância de banco de dados gravadora para seu cluster de banco de dados. No entanto, ao usar o AWS CLI para criar um cluster de banco de dados, você deve criar explicitamente a instância de banco de dados gravadora para seu cluster de banco de dados usando o `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

```

Saída:

```

{
  "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
  }
}
```

```
}
```

Exemplo 3: Para criar o grupo de fragmentos de banco de dados

O `create-db-shard-group` exemplo a seguir cria um grupo de fragmentos de banco de dados em seu cluster de banco de dados primário do 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
```

Saída:

```
{  
  "DBShardGroupResourceId": "shardgroup-a6e3a02226aa243e2ac6c7a1234567890",  
  "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"  
}
```

Para obter mais informações, consulte [Usando o Aurora Serverless v2 no Guia do usuário do Amazon Aurora](#).

- Para API obter detalhes, consulte [CreateDbShardGroup](#) na Referência de AWS CLI Comandos.

create-db-snapshot

O código de exemplo a seguir mostra como usar `create-db-snapshot`.

AWS CLI

Como criar um snapshot de banco de dados

O exemplo de `create-db-snapshot` a seguir cria um snapshot de banco de dados.

```
aws rds create-db-snapshot \  

```

```
--db-instance-identifier database-mysql \  
--db-snapshot-identifier mydbsnapshot
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Criação de um DB snapshot](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [C reateDBSnapshot](#) na Referência de AWS CLI Comandos.

create-db-subnet-group

O código de exemplo a seguir mostra como usar create-db-subnet-group.

AWS CLI

Para criar um grupo de sub-redes de banco de dados

O `create-db-subnet-group` exemplo a seguir cria um grupo de sub-redes de banco de dados chamado `mysubnetgroup` usando sub-redes existentes.

```
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"]'
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Criação de uma instância de banco de dados VPC em um](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [CreateDbSubnetGroup](#) na Referência de AWS CLI Comandos.

create-event-subscription

O código de exemplo a seguir mostra como usar `create-event-subscription`.

AWS CLI

Para criar uma assinatura de evento

O `create-event-subscription` exemplo a seguir cria uma assinatura para eventos de backup e recuperação para instâncias de banco de dados na AWS conta atual. As notificações são enviadas para um tópico do Amazon Simple Notification Service, especificado por `--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
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [CreateEventSubscription](#) na Referência de AWS CLI Comandos.

create-global-cluster

O código de exemplo a seguir mostra como usar `create-global-cluster`.

AWS CLI

Para criar um cluster de banco de dados global

O `create-global-cluster` exemplo a seguir cria um novo cluster de banco de dados global SQL compatível com o Aurora My.

```
aws rds create-global-cluster \
  --global-cluster-identifier myglobalcluster \
  --engine aurora-mysql
```

Saída:

```
{
  "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": []
  }
}
```

```
}  
}
```

Para obter mais informações, consulte [Criação de um banco de dados global do Aurora no Guia do usuário do Amazon Aurora](#).

- Para API obter detalhes, consulte [CreateGlobalCluster](#) na Referência de AWS CLI Comandos.

create-option-group

O código de exemplo a seguir mostra como usar create-option-group.

AWS CLI

Para criar um grupo de RDS opções da Amazon

O create-option-group comando a seguir cria um novo grupo de RDS opções da Amazon para a Oracle Enterprise Edition versão 11.2, is named ``MyOptionGroup e inclui uma descrição.

```
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"
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [CreateOptionGroup](#) na Referência de AWS CLI Comandos.

delete-blue-green-deployment

O código de exemplo a seguir mostra como usar delete-blue-green-deployment.

AWS CLI

Exemplo 1: Para excluir recursos em um ambiente verde RDS para uma instância for My SQL DB

O delete-blue-green-deployment exemplo a seguir exclui os recursos em um ambiente verde RDS para uma instância for My SQL DB.

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-v53303651eexfake \  
  --delete-target
```

Saída:

```
{  
  "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"
}
}

```

Para obter mais informações, consulte [Excluir uma implantação azul/verde](#) no Guia do usuário da Amazon RDS.

Exemplo 2: Para excluir recursos em um ambiente verde para um cluster Aurora My DB SQL

O delete-blue-green-deployment exemplo a seguir exclui os recursos em um ambiente verde para um cluster Aurora SQL My DB.

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \  
  --delete-target
```

Saída:

```
{  
  "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-gwzlg",
      "Status": "AVAILABLE"
    }
  ],
  "Tasks": [
    {
      "Name": "CREATING_READ_REPLICA_OF_SOURCE",
      "Status": "COMPLETED"
    },
    {
      "Name": "DB_ENGINE_VERSION_UPGRADE",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_CUSTOM_ENDPOINTS",
      "Status": "COMPLETED"
    }
  ],
  "Status": "DELETING",
  "CreateTime": "2022-02-25T21:12:00.288000+00:00",
  "DeleteTime": "2022-02-25T22:29:11.336000+00:00"
}
}

```

Para obter mais informações, consulte [Excluir uma implantação azul/verde](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteBlueGreenDeployment](#) na Referência de AWS CLI Comandos.

delete-db-cluster-endpoint

O código de exemplo a seguir mostra como usar `delete-db-cluster-endpoint`.

AWS CLI

Para excluir um endpoint de cluster de banco de dados personalizado

O `delete-db-cluster-endpoint` exemplo a seguir exclui o endpoint do cluster de banco de dados personalizado especificado.

```
aws rds delete-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint
```

Saída:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnpxample.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"  
}
```

Para obter mais informações, consulte [Amazon Aurora Connection Management](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbClusterEndpoint](#) na Referência de AWS CLI Comandos.

delete-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `delete-db-cluster-parameter-group`.

AWS CLI

Para excluir um grupo de parâmetros do cluster de banco de dados

O `delete-db-cluster-parameter-group` exemplo a seguir exclui o grupo de parâmetros do cluster de banco de dados especificado.

```
aws rds delete-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclusterparametergroup
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de clusters de banco de dados](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

delete-db-cluster-snapshot

O código de exemplo a seguir mostra como usar `delete-db-cluster-snapshot`.

AWS CLI

Para excluir um snapshot de cluster de banco de dados

O `delete-db-cluster-snapshot` exemplo a seguir exclui o snapshot do cluster de banco de dados especificado.

```
aws rds delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifier mydbclustersnapshot
```

Saída:

```
{  
  "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
}
}

```

Para obter mais informações, consulte [Excluir um snapshot](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbClusterSnapshot](#) na Referência de AWS CLI Comandos.

delete-db-cluster

O código de exemplo a seguir mostra como usar `delete-db-cluster`.

AWS CLI

Exemplo 1: Para excluir uma instância de banco de dados em um cluster de banco de dados

O `delete-db-instance` exemplo a seguir exclui a instância de banco de dados final em um cluster de banco de dados. Você não pode excluir um cluster de banco de dados se ele contiver

instâncias de banco de dados que não estejam no estado de exclusão. Você não pode tirar um instantâneo final ao excluir uma instância de banco de dados em um cluster de banco de dados.

```
aws rds delete-db-instance \  
  --db-instance-identifier database-3
```

Saída:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "database-3",  
    "DBInstanceClass": "db.r4.large",  
    "Engine": "aurora-postgresql",  
    "DBInstanceStatus": "deleting",  
  
    ...output omitted...  
  }  
}
```

Para obter mais informações, consulte [Excluir uma instância de banco de dados em um cluster de banco de dados Aurora](#) no Guia do usuário do Amazon Aurora.

Exemplo 2: Para excluir um cluster de banco de dados

O `delete-db-cluster` exemplo a seguir exclui o cluster de banco de dados chamado `mycluster` e tira um snapshot final chamado `mycluster-final-snapshot`. O status do cluster de banco de dados está disponível enquanto o snapshot está sendo obtido. Para acompanhar o progresso da exclusão, use o `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
```

Saída:

```
{  
  "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...

}
}

```

Para obter mais informações, consulte [Aurora Clusters com uma única instância de banco de dados no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbCluster](#) na Referência de AWS CLI Comandos.

delete-db-instance-automated-backup

O código de exemplo a seguir mostra como usar delete-db-instance-automated-backup.

AWS CLI

Para excluir um backup automatizado replicado de uma região

O delete-db-instance-automated-backup exemplo a seguir exclui o backup automático com o Amazon Resource Name (ARN) especificado.

```

aws rds delete-db-instance-automated-backup \
  --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"

```

Saída:

```

{
  "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-jkib2gfg5rv7replzadausbrktni2bn4example"
}
}
```

Para obter mais informações, consulte [Excluir backups replicados no Guia RDS](#) do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteDbInstanceAutomatedBackup](#) na Referência de AWS CLI Comandos.

delete-db-instance

O código de exemplo a seguir mostra como usar `delete-db-instance`.

AWS CLI

Como excluir uma instância de banco de dados

O exemplo de `delete-db-instance` a seguir exclui a instância de banco de dados especificada depois de criar um snapshot de banco de dados final chamado `test-instance-final-snap`.

```
aws rds delete-db-instance \
  --db-instance-identifier test-instance \
```

```
--final-db-snapshot-identifier test-instance-final-snap
```

Saída:

```
{
  "DBInstance": {
    "DBInstanceIdentifier": "test-instance",
    "DBInstanceStatus": "deleting",
    ...some output truncated...
  }
}
```

- Para API obter detalhes, consulte [DeleteDBInstance](#) na Referência de AWS CLI Comandos.

delete-db-parameter-group

O código de exemplo a seguir mostra como usar `delete-db-parameter-group`.

AWS CLI

Como excluir um grupo de parâmetros de banco de dados

O exemplo de command a seguir exclui um grupo de parâmetros de banco de dados.

```
aws rds delete-db-parameter-group \  
--db-parameter-group-name mydbparametergroup
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com grupos de parâmetros de banco de dados](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteDBParameterGroup](#) na Referência de AWS CLI Comandos.

delete-db-proxy-endpoint

O código de exemplo a seguir mostra como usar `delete-db-proxy-endpoint`.

AWS CLI

Para excluir um endpoint de proxy de banco de dados para um RDS banco de dados

O `delete-db-proxy-endpoint` exemplo a seguir exclui um endpoint de proxy de banco de dados para o banco de dados de destino.

```
aws rds delete-db-proxy-endpoint \  
  --db-proxy-endpoint-name proxyEP1
```

Saída:

```
{  
  "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  
    }  
}
```

Para obter mais informações, consulte [Excluir um endpoint proxy](#) no Guia RDS do usuário da Amazon e [Excluir um endpoint proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbProxyEndpoint](#) na Referência de AWS CLI Comandos.

delete-db-proxy

O código de exemplo a seguir mostra como usar `delete-db-proxy`.

AWS CLI

Para excluir um proxy de banco de dados para um RDS banco de dados

O `delete-db-proxy` exemplo a seguir exclui um proxy de banco de dados.

```
aws rds delete-db-proxy \  
  --db-proxy-name proxyExample
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Excluindo um RDS proxy](#) no Guia do RDSusuário da Amazon e [Excluindo um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbProxy](#) na Referência de AWS CLI Comandos.

delete-db-security-group

O código de exemplo a seguir mostra como usar delete-db-security-group.

AWS CLI

Para excluir um grupo de segurança de banco de dados

O delete-db-security-group exemplo a seguir exclui um grupo de segurança de banco de dados chamado mysecuritygroup.

```
aws rds delete-db-security-group \  
  --db-security-group-name mysecuritygroup
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com grupos de segurança de banco de dados \(plataforma EC2 -Classic\)](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [DeleteDbSecurityGroup](#) na Referência de AWS CLI Comandos.

delete-db-shard-group

O código de exemplo a seguir mostra como usar delete-db-shard-group.

AWS CLI

Exemplo 1: Para excluir um grupo de fragmentos de banco de dados sem sucesso

O delete-db-shard-group exemplo a seguir mostra o erro que ocorre quando você tenta excluir um grupo de fragmentos de banco de dados antes de excluir todos os seus bancos de dados e esquemas.

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifier limitless-test-shard-grp
```

Saída:

```
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.
```

Exemplo 2: Para excluir um grupo de fragmentos de banco de dados com sucesso

O delete-db-shard-group exemplo a seguir exclui um grupo de fragmentos de banco de dados depois que você exclui todos os seus bancos de dados e esquemas, incluindo o esquema public

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifier limitless-test-shard-grp
```

Saída:

```
{  
  "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-cekyexample.us-  
east-2.rds.amazonaws.com"  
}
```

Para obter mais informações, consulte [Excluir clusters e instâncias de banco de dados do Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteDbShardGroup](#) na Referência de AWS CLI Comandos.

delete-db-snapshot

O código de exemplo a seguir mostra como usar delete-db-snapshot.

AWS CLI

Para excluir um DB snapshot

O `delete-db-snapshot` exemplo a seguir exclui o DB snapshot especificado.

```
aws rds delete-db-snapshot \  
  --db-snapshot-identifier mydbsnapshot
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Excluindo um snapshot no Guia RDS](#) do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteDbSnapshot](#) na Referência de AWS CLI Comandos.

delete-db-subnet-group

O código de exemplo a seguir mostra como usar `delete-db-subnet-group`.

AWS CLI

Para excluir um grupo de sub-redes de banco de dados

O `delete-db-subnet-group` exemplo a seguir exclui o grupo de sub-rede de banco de dados chamado `mysubnetgroup`

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com uma instância de banco de dados VPC em um](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DeleteDbSubnetGroup](#) na Referência de AWS CLI Comandos.

delete-event-subscription

O código de exemplo a seguir mostra como usar `delete-event-subscription`.

AWS CLI

Para excluir uma assinatura de evento

O `delete-event-subscription` exemplo a seguir exclui a assinatura do evento especificada.

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

Saída:

```
{  
  "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"
}
}

```

- Para API obter detalhes, consulte [DeleteEventSubscription](#) na Referência de AWS CLI Comandos.

delete-global-cluster

O código de exemplo a seguir mostra como usar `delete-global-cluster`.

AWS CLI

Para excluir um cluster de banco de dados global

O `delete-global-cluster` exemplo a seguir exclui um cluster de banco de dados global compatível com o Aurora SQL My. A saída mostra o cluster que você está excluindo, mas os `describe-global-clusters` comandos subsequentes não listam esse cluster de banco de dados.

```

aws rds delete-global-cluster \
  --global-cluster-identifier myglobalcluster

```

Saída:

```
{
```

```
"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": []
}
```

Para obter mais informações, consulte [Excluir um banco de dados global do Aurora](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeleteGlobalCluster](#) na Referência de AWS CLI Comandos.

delete-option-group

O código de exemplo a seguir mostra como usar delete-option-group.

AWS CLI

Para excluir um grupo de opções

O delete-option-group exemplo a seguir exclui o grupo de opções especificado.

```
aws rds delete-option-group \
  --option-group-name myoptiongroup
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um grupo de opções](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [DeleteOptionGroup](#) na Referência de AWS CLI Comandos.

deregister-db-proxy-targets

O código de exemplo a seguir mostra como usar deregister-db-proxy-targets.

AWS CLI

Para cancelar o registro de um alvo de proxy de banco de dados do grupo de destino do banco de dados

O `deregister-db-proxy-targets` exemplo a seguir remove a associação entre o proxy `proxyExample` e seu destino.

```
aws rds deregister-db-proxy-targets \  
  --db-proxy-name proxyExample \  
  --db-instance-identifiers database-1
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo um RDS proxy](#) no Guia do RDS usuário da Amazon e [Excluindo um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DeregisterDbProxyTargets](#) na Referência de AWS CLI Comandos.

`describe-account-attributes`

O código de exemplo a seguir mostra como usar `describe-account-attributes`.

AWS CLI

Como descrever os atributos da conta

O `describe-account-attributes` exemplo a seguir recupera os atributos da AWS conta atual.

```
aws rds describe-account-attributes
```

Saída:

```
{  
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeAccountAttributes](#) na Referência de AWS CLI Comandos.

describe-blue-green-deployments

O código de exemplo a seguir mostra como usar `describe-blue-green-deployments`.

AWS CLI

Exemplo 1: Para descrever uma implantação azul/verde de uma RDS instância de banco de dados após a conclusão da criação

O `describe-blue-green-deployment` exemplo a seguir recupera os detalhes de uma implantação azul/verde após a conclusão da criação.

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-v53303651eexfake
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Visualização de uma implantação azul/verde no Guia RDS](#) do usuário da Amazon.

Exemplo 2: Para descrever uma implantação azul/verde para um cluster Aurora My DB SQL

O `describe-blue-green-deployment` exemplo a seguir recupera os detalhes de uma implantação azul/verde.

```

aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake

```

Saída:

```

{
  "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"
}
]
}

```

Para obter mais informações, consulte [Visualização de uma implantação azul/verde no Guia](#) do usuário do Amazon Aurora.

Exemplo 3: Para descrever uma implantação azul/verde para um cluster Aurora My após a transição SQL

O `describe-blue-green-deployment` exemplo a seguir recupera os detalhes sobre uma implantação azul/verde depois que o ambiente verde é promovido para ser o ambiente de produção.

```

aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzlgccsfake

```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Visualização de uma implantação azul/verde no Guia](#) do usuário do Amazon Aurora.

Exemplo 4: Para descrever uma implantação combinada azul/verde

O `describe-blue-green-deployment` exemplo a seguir recupera os detalhes de uma implantação combinada azul/verde.

```
aws rds describe-blue-green-deployments
```

Saída:

```
{
  "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"
```

```
    }  
  ]  
}
```

Para obter mais informações, consulte [Visualização de uma implantação azul/verde](#) no Guia RDS do usuário da Amazon e [Visualização de uma implantação azul/verde](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeBlueGreenDeployments](#) na Referência de AWS CLI Comandos.

describe-certificates

O código de exemplo a seguir mostra como usar `describe-certificates`.

AWS CLI

Para descrever certificados

O `describe-certificates` exemplo a seguir recupera os detalhes do certificado associado à região padrão do usuário.

```
aws rds describe-certificates
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

```

    "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"
}

```

Para obter mais informações, consulte [UsandoSSL/TLSpara criptografar uma conexão com uma instância](#) de banco de dados no Amazon RDS User Guide e [UsandoSSL/TLSpara criptografar uma conexão com um cluster de banco](#) de dados no Amazon Aurora User Guide.

- Para API obter detalhes, consulte [DescribeCertificates](#) na Referência de AWS CLI Comandos.

describe-db-cluster-backtracks

O código de exemplo a seguir mostra como usar `describe-db-cluster-backtracks`.

AWS CLI

Para descrever os retrocessos de um cluster de banco de dados

O `describe-db-cluster-backtracks` exemplo a seguir recupera detalhes sobre o cluster de banco de dados especificado.


```
aws rds describe-db-cluster-backtracks \  
--db-cluster-identifier mydbcluster
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Backtracking an Aurora DB](#) cluster no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusterBacktracks](#) na Referência de AWS CLI Comandos.

describe-db-cluster-endpoints

O código de exemplo a seguir mostra como usar `describe-db-cluster-endpoints`.

AWS CLI

Exemplo 1: Para descrever os endpoints do cluster de banco de dados

O `describe-db-cluster-endpoints` exemplo a seguir recupera detalhes dos endpoints do seu cluster de banco de dados. Os tipos mais comuns de clusters do Aurora têm dois endpoints.

Um endpoint tem tipoWRITER. Você pode usar esse endpoint para todas as SQL declarações. O outro endpoint tem tipoREADER. Você pode usar esse endpoint somente para SELECT e outras instruções somente para leituraSQL.

```
aws rds describe-db-cluster-endpoints
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: Para descrever os endpoints do cluster de banco de dados de um único cluster de banco de dados

O `describe-db-cluster-endpoints` exemplo a seguir recupera detalhes dos endpoints do cluster de banco de dados de um único cluster de banco de dados especificado. Os clusters sem servidor do Aurora têm apenas um único endpoint com um tipo de `WRITER`

```
aws rds describe-db-cluster-endpoints \
  --db-cluster-identifier serverless-cluster
```

Saída:

```
{
  "DBClusterEndpoints": [
    {
      "Status": "available",
      "Endpoint": "serverless-cluster.cluster-cnpxample.us-
east-1.rds.amazonaws.com",
      "DBClusterIdentifier": "serverless-cluster",
      "EndpointType": "WRITER"
    }
  ]
}
```

Para obter mais informações, consulte [Amazon Aurora Connection Management](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusterEndpoints](#) na Referência de AWS CLI Comandos.

describe-db-cluster-parameter-groups

O código de exemplo a seguir mostra como usar `describe-db-cluster-parameter-groups`.

AWS CLI

Para descrever grupos de parâmetros do cluster de banco de dados

O `describe-db-cluster-parameter-groups` exemplo a seguir recupera detalhes dos grupos de parâmetros do seu cluster de banco de dados.

```
aws rds describe-db-cluster-parameter-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de clusters de banco de dados](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusterParameterGroups](#) na Referência de AWS CLI Comandos.

describe-db-cluster-parameters

O código de exemplo a seguir mostra como usar `describe-db-cluster-parameters`.

AWS CLI

Exemplo 1: Para descrever os parâmetros em um grupo de parâmetros de cluster de banco de dados

O `describe-db-cluster-parameters` exemplo a seguir recupera detalhes sobre os parâmetros em um grupo de parâmetros de cluster de banco de dados.

```
aws rds describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name mydbclusterpg
```

Saída:

```
{  
  "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...
]
}

```

Exemplo 2: Para listar somente os nomes dos parâmetros em um grupo de parâmetros do cluster de banco de dados

O `describe-db-cluster-parameters` exemplo a seguir recupera somente os nomes dos parâmetros em um grupo de parâmetros do cluster de banco de dados.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].{ParameterName:ParameterName}'

```

Saída:

```

[
  {
    "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...
}
]

```

Exemplo 3: Para descrever somente os parâmetros modificáveis em um grupo de parâmetros do cluster de banco de dados

O `describe-db-cluster-parameters` exemplo a seguir recupera os nomes somente dos parâmetros que você pode modificar em um grupo de parâmetros do cluster de banco de dados.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].[ParameterName:ParameterName,IsModifiable:IsModifiable] | [?IsModifiable == `true`]'

```

Saída:

```

[
  {
    "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...
}
]

```

Exemplo 4: Para descrever somente os parâmetros booleanos modificáveis em um grupo de parâmetros do cluster de banco de dados

O `describe-db-cluster-parameters` exemplo a seguir recupera somente os nomes dos parâmetros que você pode modificar em um grupo de parâmetros de cluster de banco de dados e que têm um tipo de dados 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`]'
```

Saída:

```
[  
  {  
    "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...  
]
```


Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de clusters de banco de dados](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusterParameters](#) na Referência de AWS CLI Comandos.

describe-db-cluster-snapshot-attributes

O código de exemplo a seguir mostra como usar `describe-db-cluster-snapshot-attributes`.

AWS CLI

Para descrever os nomes e valores dos atributos de um snapshot de cluster de banco de dados

O `describe-db-cluster-snapshot-attributes` exemplo a seguir recupera detalhes dos nomes e valores dos atributos para o snapshot do cluster de banco de dados especificado.

```
aws rds describe-db-cluster-snapshot-attributes \  
--db-cluster-snapshot-identifier myclustersnapshot
```

Saída:

```
{  
  "DBClusterSnapshotAttributesResult": {  
    "DBClusterSnapshotIdentifier": "myclustersnapshot",  
    "DBClusterSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789012"  
        ]  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Compartilhamento de um snapshot de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusterSnapshotAttributes](#) na Referência de AWS CLI Comandos.

describe-db-cluster-snapshots

O código de exemplo a seguir mostra como usar `describe-db-cluster-snapshots`.

AWS CLI

Para descrever um snapshot de cluster de banco de dados para um cluster de banco de dados

O `describe-db-cluster-snapshots` exemplo a seguir recupera os detalhes dos snapshots do cluster de banco de dados para o cluster de banco de dados especificado.

```
aws rds describe-db-cluster-snapshots \  
--db-cluster-identifier mydbcluster
```

Saída:

```
{  
  "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
  }
]
}

```

Para obter mais informações, consulte [Criação de um snapshot de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusterSnapshots](#) na Referência de AWS CLI Comandos.

describe-db-clusters

O código de exemplo a seguir mostra como usar `describe-db-clusters`.

AWS CLI

Exemplo 1: Para descrever um cluster de banco de dados

O `describe-db-clusters` exemplo a seguir recupera os detalhes do cluster de banco de dados especificado.

```
aws rds describe-db-clusters \  
--db-cluster-identifier mydbcluster
```

Saída:

```
{  
  "DBClusters": [  
    {  
      "AllocatedStorage": 1,  
      "AvailabilityZones": [  
        "us-east-1a",  
        "us-east-1b",  
        "us-east-1e"  
      ],  
      "BackupRetentionPeriod": 1,  
      "DatabaseName": "mydbcluster",  
      "DBClusterIdentifier": "mydbcluster",  
      "DBClusterParameterGroup": "default.aurora-mysql5.7",  
      "DBSubnetGroup": "default",  
      "Status": "available",  
      "EarliestRestorableTime": "2019-06-19T09:16:28.210Z",  
      "Endpoint": "mydbcluster.cluster-cnpxample.us-  
east-1.rds.amazonaws.com",  
      "ReaderEndpoint": "mydbcluster.cluster-ro-cnpxample.us-  
east-1.rds.amazonaws.com",  
      "MultiAZ": true,  
      "Engine": "aurora-mysql",  
      "EngineVersion": "5.7.mysql_aurora.2.04.2",  
      "LatestRestorableTime": "2019-06-20T22:38:14.908Z",  
      "Port": 3306,  
      "MasterUsername": "myadmin",  
      "PreferredBackupWindow": "09:09-09:39",
```

```
"PreferredMaintenanceWindow": "sat:04:09-sat:04:39",
"ReadReplicaIdentifiers": [],
"DBClusterMembers": [
  {
    "DBInstanceIdentifier": "dbinstance3",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "dbinstance1",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "dbinstance2",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "mydbcluster",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "mydbcluster-us-east-1b",
    "IsClusterWriter": false,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  },
  {
    "DBInstanceIdentifier": "mydbcluster",
    "IsClusterWriter": true,
    "DBClusterParameterGroupStatus": "in-sync",
    "PromotionTier": 1
  }
],
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-0b9130572daf3dc16",
    "Status": "active"
  }
]
```

```

    }
  ],
  "HostedZoneId": "Z2R2ITUGPM61AM",
  "StorageEncrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:814387698303:key/
AKIAIOSFODNN7EXAMPLE",
  "DbClusterResourceId": "cluster-AKIAIOSFODNN7EXAMPLE",
  "DBClusterArn": "arn:aws:rds:us-
east-1:123456789012:cluster:mysqlcluster",
  "AssociatedRoles": [],
  "IAMDatabaseAuthenticationEnabled": false,
  "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
  "EngineMode": "provisioned",
  "DeletionProtection": false,
  "HttpEndpointEnabled": false
}
]
}

```

Exemplo 2: Para listar determinados atributos de todos os clusters de banco de dados

O `describe-db-clusters` exemplo a seguir recupera somente os `ReaderEndpoint` atributos `DBClusterIdentifier`, `Endpoint`, e de todos os seus clusters de banco de dados na AWS região atual.

```

aws rds describe-db-clusters \
  --query 'DBClusters[.
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}]'

```

Saída:

```

[
  {
    "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpxexample.us-
east-1.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpxexample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "cluster-57-2020-05-01-2270"
  },
  {
    "Endpoint": "cluster-57-2020-05-01-4615.cluster-cnpxexample.us-
east-1.rds.amazonaws.com",

```

```

    "ReaderEndpoint": "cluster-57-2020-05-01-4615.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "cluster-57-2020-05-01-4615"
  },
  {
    "Endpoint": "pg2-cluster.cluster-cnpxample.us-east-1.rds.amazonaws.com",
    "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "pg2-cluster"
  },
  ...output omitted...
}
]

```

Exemplo 3: Para listar clusters de banco de dados com um atributo específico

O `describe-db-clusters` exemplo a seguir recupera somente Engine os atributos `DBClusterIdentifier` e dos seus clusters de banco de dados que usam o mecanismo de `aurora-postgresql` banco de dados.

```

aws rds describe-db-clusters \
  --query 'DBClusters[].[DBClusterIdentifier:DBClusterIdentifier,Engine:Engine] |
  [?Engine == `aurora-postgresql`]'

```

Saída:

```

[
  {
    "Engine": "aurora-postgresql",
    "DBClusterIdentifier": "pg2-cluster"
  }
]

```

Para obter mais informações, consulte [Amazon Aurora DB Clusters](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbClusters](#) na Referência de AWS CLI Comandos.

describe-db-engine-versions

O código de exemplo a seguir mostra como usar `describe-db-engine-versions`.

AWS CLI

Para descrever as versões do mecanismo de banco de dados para o mecanismo My SQL DB

O exemplo de `describe-db-engine-versions` a seguir exibe os detalhes de cada uma das versões do mecanismo de banco de dados para o mecanismo de banco de dados especificado.

```
aws rds describe-db-engine-versions \  
  --engine mysql
```

Saída:

```
{  
  "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...  
      ]  
    }  
  ]  
}
```



```
]
}
```

Para obter mais informações, consulte [O que é o Amazon Relational Database Service \(RDSAmazon\)](#)? no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [escribeDBEngineVersões D](#) na Referência de AWS CLI Comandos.

describe-db-instance-automated-backups

O código de exemplo a seguir mostra como usar `describe-db-instance-automated-backups`.

AWS CLI

Para descrever os backups automatizados para uma instância de banco de dados

O `describe-db-instance-automated-backups` exemplo a seguir exibe detalhes sobre os backups automatizados para a instância de banco de dados especificada. Os detalhes incluem backups automatizados replicados em outras AWS regiões.

```
aws rds describe-db-instance-automated-backups \
  --db-instance-identifier new-orcl-db
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte [Encontrar informações sobre backups replicados](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [DescribeDbInstanceAutomatedBackups](#) na Referência de AWS CLI Comandos.

describe-db-instances

O código de exemplo a seguir mostra como usar `describe-db-instances`.

AWS CLI

Como descrever uma instância de banco de dados

O exemplo de `describe-db-instances` a seguir recupera detalhes sobre a instância de banco de dados especificada.

```

aws rds describe-db-instances \
  --db-instance-identifier mydbinstancecf

```

Saída:

```

{
  "DBInstances": [
    {
      "DBInstanceIdentifier": "mydbinstancecf",
      "DBInstanceClass": "db.t3.small",
      "Engine": "mysql",

```

```

        "DBInstanceStatus": "available",
        "MasterUsername": "masterawsuser",
        "Endpoint": {
            "Address": "mydbinstancecf.abcxample.us-east-1.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        ...some output truncated...
    }
]
}

```

- Para API obter detalhes, consulte [Descreva as instâncias de banco de dados](#) na Referência de AWS CLI Comandos.

describe-db-log-files

O código de exemplo a seguir mostra como usar `describe-db-log-files`.

AWS CLI

Para descrever os arquivos de log de uma instância de banco de dados

O `describe-db-log-files` exemplo a seguir recupera detalhes sobre os arquivos de log da instância de banco de dados especificada.

```

aws rds describe-db-log-files -\
  -db-instance-identifier test-instance

```

Saída:

```

{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeDbLogFiles](#) na Referência de AWS CLI Comandos.

describe-db-parameter-groups

O código de exemplo a seguir mostra como usar `describe-db-parameter-groups`.

AWS CLI

Como descrever um grupo de parâmetros de banco de dados

O exemplo de `describe-db-parameter-groups` a seguir recupera detalhes sobre os grupos de parâmetros de banco de dados.

```
aws rds describe-db-parameter-groups
```

Saída:

```
{
```

```
"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...  
]  
}
```

Para obter mais informações, consulte [Trabalhando com grupos de parâmetros de banco de dados](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [escribeDBParameterGrupos de D](#) na Referência de AWS CLI Comandos.

describe-db-parameters

O código de exemplo a seguir mostra como usar `describe-db-parameters`.

AWS CLI

Como descrever os parâmetros de um grupo de parâmetros de banco de dados

O exemplo de `describe-db-parameters` a seguir recupera detalhes sobre o grupo de parâmetros de banco de dados especificado.

```
aws rds describe-db-parameters \  
  --db-parameter-group-name mydbpg
```

Saída:

```
{  
  "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...  
  ]  
}
```

Para obter mais informações, consulte [Trabalhando com grupos de parâmetros de banco de dados](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [D `describeDBParameters`](#) na Referência de AWS CLI Comandos.

`describe-db-proxies`

O código de exemplo a seguir mostra como usar `describe-db-proxies`.

AWS CLI

Para descrever um proxy de banco de dados para um RDS banco de dados

O `describe-db-proxies` exemplo a seguir retorna informações sobre proxies de banco de dados.

```
aws rds describe-db-proxies
```

Saída:

```
{
  "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",
    "UpdateDate": "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",
    "UpdateDate": "2023-04-13T01:49:38.568000+00:00"
  }
]

```



```
}
```

Para obter mais informações, consulte [Visualização de um RDS proxy](#) no Guia do RDS usuário da Amazon e [Visualização de um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbProxies](#) na Referência de AWS CLI Comandos.

describe-db-proxy-endpoints

O código de exemplo a seguir mostra como usar `describe-db-proxy-endpoints`.

AWS CLI

Para descrever os endpoints de um proxy de banco de dados

O `describe-db-proxy-endpoints` exemplo a seguir retorna informações sobre endpoints de proxy de banco de dados.

```
aws rds describe-db-proxy-endpoints
```

Saída:

```
{
  "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
    }
]
}

```

Para obter mais informações, consulte [Visualização de um endpoint proxy](#) no Guia do RDS usuário da Amazon e [Criação de um endpoint proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbProxyEndpoints](#) na Referência de AWS CLI Comandos.

describe-db-proxy-target-groups

O código de exemplo a seguir mostra como usar `describe-db-proxy-target-groups`.

AWS CLI

Para descrever os endpoints de um proxy de banco de dados

O `describe-db-proxy-target-groups` exemplo a seguir retorna informações sobre grupos de destino de proxy de banco de dados.

```
aws rds describe-db-proxy-target-groups \
```

```
--db-proxy-name proxyExample
```

Saída:

```
{
  "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"
    }
}
```

Para obter mais informações, consulte [Visualização de um RDS proxy](#) no Guia do RDS usuário da Amazon e [Visualização de um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbProxyTargetGroups](#) na Referência de AWS CLI Comandos.

describe-db-proxy-targets

O código de exemplo a seguir mostra como usar `describe-db-proxy-targets`.

AWS CLI

Para descrever os alvos do proxy de banco de dados

O `describe-db-proxy-targets` exemplo a seguir retorna informações sobre destinos de proxy de banco de dados.

```
aws rds describe-db-proxy-targets \
```

```
--db-proxy-name proxyExample
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Visualização de um RDS proxy](#) no Guia do RDS usuário da Amazon e [Visualização de um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbProxyTargets](#) na Referência de AWS CLI Comandos.

describe-db-recommendations

O código de exemplo a seguir mostra como usar `describe-db-recommendations`.

AWS CLI

Exemplo 1: Para listar todas as recomendações de banco de dados

O `describe-db-recommendations` exemplo a seguir lista todas as recomendações de banco de dados em sua AWS conta.

```
aws rds describe-db-recommendations
```

Saída:

```
{
  "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"
    }
  ]
}
]
}

```

Para obter mais informações, consulte [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do RDS usuário da Amazon e [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do usuário do Amazon Aurora.

Exemplo 2: Para listar recomendações de banco de dados de alta severidade

O `describe-db-recommendations` exemplo a seguir lista recomendações de banco de dados de alta severidade em sua AWS conta.

```
aws rds describe-db-recommendations \  
  --filters Name=severity,Values=high
```

Saída:

```
{  
  "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.\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/"
      }
    ]
  }
]
```


Para obter mais informações, consulte [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do RDS usuário da Amazon e [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do usuário do Amazon Aurora.

Exemplo 3: Para listar recomendações de banco de dados para uma instância de banco de dados especificada

O `describe-db-recommendations` exemplo a seguir lista todas as recomendações de banco de dados para uma instância de banco de dados especificada.

```
aws rds describe-db-recommendations \
  --filters Name=dbi-resource-id,Values=database-1
```

Saída:

```
{
  "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"
    }
]
}
}
}

```

Para obter mais informações, consulte [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do RDS usuário da Amazon e [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do usuário do Amazon Aurora.

Exemplo 4: Para listar todas as recomendações ativas do banco de dados

O `describe-db-recommendations` exemplo a seguir lista todas as recomendações ativas de banco de dados em sua AWS conta.

```

aws rds describe-db-recommendations \
  --filters Name=status,Values=active

```

Saída:

```

{
  "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"
    }
  ]
}
]
```

Para obter mais informações, consulte [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do RDS usuário da Amazon e [Visualizar e responder às RDS recomendações da Amazon](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbRecommendations](#) na Referência de AWS CLI Comandos.

describe-db-security-groups

O código de exemplo a seguir mostra como usar `describe-db-security-groups`.

AWS CLI

Para listar grupos de segurança de banco de dados

O `describe-db-security-groups` exemplo a seguir lista os grupos de segurança do banco de dados.

```
aws rds describe-db-security-groups
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações, consulte [Listar grupos de segurança de banco de dados disponíveis](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeDbSecurityGroups](#) na Referência de AWS CLI Comandos.

describe-db-shard-groups

O código de exemplo a seguir mostra como usar `describe-db-shard-groups`.

AWS CLI

Exemplo 1: Para descrever grupos de fragmentos de banco de dados

O `describe-db-shard-groups` exemplo a seguir recupera os detalhes dos seus grupos de fragmentos de banco de dados.

```
aws rds describe-db-shard-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Amazon Aurora DB Clusters](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeDbShardGroups](#) na Referência de AWS CLI Comandos.

describe-db-snapshot-attributes

O código de exemplo a seguir mostra como usar `describe-db-snapshot-attributes`.

AWS CLI

Para descrever os nomes e valores dos atributos de um DB snapshot

O `describe-db-snapshot-attributes` exemplo a seguir descreve os nomes e valores dos atributos de um DB snapshot.

```
aws rds describe-db-snapshot-attributes \  
  --db-snapshot-identifier mydbsnapshot
```

Saída:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "123456789012",  
          "210987654321"  
        ]  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Compartilhamento de um DB snapshot](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [DescribeDbSnapshotAttributes](#) na Referência de AWS CLI Comandos.

describe-db-snapshots

O código de exemplo a seguir mostra como usar `describe-db-snapshots`.

AWS CLI

Exemplo 1: como descrever um snapshot de banco de dados de uma instância de banco de dados

O exemplo de `describe-db-snapshots` a seguir recupera os detalhes de um snapshot de banco de dados para uma instância de banco de dados.

```
aws rds describe-db-snapshots \  
  --db-snapshot-identifier mydbsnapshot
```


Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Criação de um DB snapshot](#) no Guia do RDS usuário da Amazon.

Exemplo 2: como encontrar o número de snapshots manuais criados

O `describe-db-snapshots` exemplo a seguir usa o `length` operador na `--query` opção para retornar o número de instantâneos manuais que foram tirados em uma AWS região específica.

```
aws rds describe-db-snapshots \
```

```
--snapshot-type manual \
--query "length(*[].{DBSnapshots:SnapshotType})" \
--region eu-central-1
```

Saída:

```
35
```

Para obter mais informações, consulte [Criação de um DB snapshot](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [D escribeDBSnapshots](#) na Referência de AWS CLI Comandos.

describe-db-subnet-groups

O código de exemplo a seguir mostra como usar `describe-db-subnet-groups`.

AWS CLI

Para descrever um grupo de sub-redes de banco de dados

O `describe-db-subnet-groups` exemplo a seguir recupera os detalhes do grupo de sub-rede de banco de dados especificado.

```
aws rds describe-db-subnet-groups
```

Saída:

```
{
  "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"
}
]
}

```

Para obter mais informações, consulte [Amazon Virtual Private Cloud VPCs e Amazon RDS](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeDbSubnetGroups](#) na Referência de AWS CLI Comandos.

describe-engine-default-cluster-parameters

O código de exemplo a seguir mostra como usar `describe-engine-default-cluster-parameters`.

AWS CLI

Para descrever o mecanismo padrão e as informações de parâmetros do sistema para o mecanismo de banco de dados Aurora

O `describe-engine-default-cluster-parameters` exemplo a seguir recupera os detalhes do mecanismo padrão e as informações dos parâmetros do sistema para clusters de banco de dados Aurora com compatibilidade com o SQL My 5.7.

```
aws rds describe-engine-default-cluster-parameters \  
--db-parameter-group-family aurora-mysql5.7
```

Saída:

```
{  
  "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...  
    ]  
  }  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de clusters de banco de dados](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeEngineDefaultClusterParameters](#) na Referência de AWS CLI Comandos.

describe-engine-default-parameters

O código de exemplo a seguir mostra como usar `describe-engine-default-parameters`.

AWS CLI

Para descrever o mecanismo padrão e as informações de parâmetros do sistema para o mecanismo de banco de dados

O `describe-engine-default-parameters` exemplo a seguir recupera detalhes do mecanismo padrão e das informações dos parâmetros do sistema para as instâncias de banco de dados My SQL 5.7.

```
aws rds describe-engine-default-parameters \  
  --db-parameter-group-family mysql5.7
```

Saída:

```
{  
  "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...  
    ]  
  }  
}
```

Para obter mais informações, consulte [Trabalhando com grupos de parâmetros de banco de dados](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeEngineDefaultParameters](#) na Referência de AWS CLI Comandos.

describe-event-categories

O código de exemplo a seguir mostra como usar `describe-event-categories`.

AWS CLI

Para descrever categorias de eventos

O `describe-event-categories` exemplo a seguir recupera detalhes sobre as categorias de eventos para todas as fontes de eventos disponíveis.

```
aws rds describe-event-categories
```

Saída:

```
{
  "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"
  ]
}
]
```

- Para API obter detalhes, consulte [DescribeEventCategories](#) na Referência de AWS CLI Comandos.

describe-event-subscriptions

O código de exemplo a seguir mostra como usar `describe-event-subscriptions`.

AWS CLI

Para descrever as assinaturas de eventos

Este exemplo descreve todas as assinaturas de RDS eventos da Amazon para a conta corrente AWS .

```
aws rds describe-event-subscriptions
```

Saída:

```
{
  "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...
  ]
}
```

- Para API obter detalhes, consulte [DescribeEventSubscriptions](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Para descrever eventos

O `describe-events` exemplo a seguir recupera detalhes dos eventos que ocorreram na instância de banco de dados especificada.


```
aws rds describe-events \  
  --source-identifier test-instance \  
  --source-type db-instance
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-export-tasks

O código de exemplo a seguir mostra como usar `describe-export-tasks`.

AWS CLI

Para descrever as tarefas de exportação de instantâneos

O `describe-export-tasks` exemplo a seguir retorna informações sobre exportações de snapshots para o Amazon S3.

```
aws rds describe-export-tasks
```

Saída:

```
{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Monitoramento de exportações de snapshots](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [DescribeExportTasks](#) na Referência de AWS CLI Comandos.

describe-global-clusters

O código de exemplo a seguir mostra como usar `describe-global-clusters`.

AWS CLI

Para descrever clusters de banco de dados globais

O `describe-global-clusters` exemplo a seguir lista os clusters de banco de dados globais do Aurora na região atual AWS .

```
aws rds describe-global-clusters
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciamento de um banco de dados global do Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [DescribeGlobalClusters](#) na Referência de AWS CLI Comandos.

describe-option-group-options

O código de exemplo a seguir mostra como usar `describe-option-group-options`.

AWS CLI

Para descrever todas as opções disponíveis

O `describe-option-group-options` exemplo a seguir lista duas opções para uma instância do Oracle Database 19c.

```
aws rds describe-option-group-options \  
  --engine-name oracle-ee \  
  --major-engine-version 19 \  
  --max-items 2
```

Saída:

```
{  
  "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=="
}

```

Para obter mais informações, consulte [Listar as opções e as configurações de opções para um grupo de opções](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeOptionGroupOptions](#) na Referência de AWS CLI Comandos.

describe-option-groups

O código de exemplo a seguir mostra como usar describe-option-groups.

AWS CLI

Para descrever os grupos de opções disponíveis

O describe-option-groups exemplo a seguir lista os grupos de opções para uma instância do Oracle Database 19c.

```

aws rds describe-option-groups \
  --engine-name oracle-ee \

```

```
--major-engine-version 19
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Listar as opções e as configurações de opções para um grupo de opções](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeOptionGroups](#) na Referência de AWS CLI Comandos.

describe-orderable-db-instance-options

O código de exemplo a seguir mostra como usar `describe-orderable-db-instance-options`.

AWS CLI

Como descrever as opções de instância solicitável do banco de dados

O `describe-orderable-db-instance-options` exemplo a seguir recupera detalhes sobre as opções solicitáveis para uma instância de banco de dados executando o mecanismo My SQL DB.

```
aws rds describe-orderable-db-instance-options \
  --engine mysql
```

Saída:

```
{
  "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...
}
```

- Para API obter detalhes, consulte [DescribeOrderableDBInstanceOptions](#) na Referência de AWS CLI Comandos.

describe-pending-maintenance-actions

O código de exemplo a seguir mostra como usar `describe-pending-maintenance-actions`.

AWS CLI

Para listar recursos com pelo menos uma ação de manutenção pendente

O `describe-pending-maintenance-actions` exemplo a seguir lista a ação de manutenção pendente para uma instância de banco de dados.

```
aws rds describe-pending-maintenance-actions
```

Saída:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [Manutenção de uma instância](#) de banco de dados no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribePendingMaintenanceActions](#) na Referência de AWS CLI Comandos.

describe-reserved-db-instances-offerings

O código de exemplo a seguir mostra como usar `describe-reserved-db-instances-offerings`.

AWS CLI

Para descrever as ofertas de instâncias de banco de dados reservadas

O `describe-reserved-db-instances-offerings` exemplo a seguir recupera detalhes sobre as opções de instância de banco de dados reservada para `oracle`.

```
aws rds describe-reserved-db-instances-offerings \  
  --product-description oracle
```

Saída:

```
{  
  "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...  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeReservedDbInstancesOfferings](#) na Referência de AWS CLI Comandos.

`describe-reserved-db-instances`

O código de exemplo a seguir mostra como usar `describe-reserved-db-instances`.

AWS CLI

Para descrever instâncias de banco de dados reservadas

O `describe-reserved-db-instances` exemplo a seguir recupera detalhes sobre todas as instâncias de banco de dados reservadas na AWS conta atual.

```
aws rds describe-reserved-db-instances
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Instâncias de banco de dados reservadas para a Amazon RDS](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [DescribeReservedDbInstances](#) na Referência de AWS CLI Comandos.

describe-source-regions

O código de exemplo a seguir mostra como usar `describe-source-regions`.

AWS CLI

Para descrever as regiões de origem

O `describe-source-regions` exemplo a seguir recupera detalhes sobre todas as AWS regiões de origem. Também mostra que os backups automatizados só podem ser replicados do Oeste dos EUA (Oregon) para a AWS região de destino, Leste dos EUA (Norte da Virgínia).

```
aws rds describe-source-regions \  
  --region us-east-1
```

Saída:

```
{  
  "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
}
]
```

Para obter mais informações, consulte [Encontrar informações sobre backups replicados](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [DescribeSourceRegions](#) na Referência de AWS CLI Comandos.

describe-valid-db-instance-modifications

O código de exemplo a seguir mostra como usar `describe-valid-db-instance-modifications`.

AWS CLI

Para descrever modificações válidas para uma instância de banco de dados

O `describe-valid-db-instance-modifications` exemplo a seguir recupera detalhes sobre as modificações válidas para a instância de banco de dados especificada.

```
aws rds describe-valid-db-instance-modifications \
  --db-instance-identifier test-instance
```

Saída:

```
{
  "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"
}
]
```



```
}
```

- Para API obter detalhes, consulte [DescribeValidDbInstanceModifications](#) na Referência de AWS CLI Comandos.

download-db-log-file-portion

O código de exemplo a seguir mostra como usar `download-db-log-file-portion`.

AWS CLI

Para baixar um arquivo de log do banco de dados

O `download-db-log-file-portion` exemplo a seguir baixa somente a parte mais recente do seu arquivo de log, salvando-a em um arquivo local chamado `tail.txt`.

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier test-instance \  
  --log-file-name log.txt \  
  --output text > tail.txt
```

Para baixar o arquivo inteiro, você precisa incluir o `--starting-token 0` parâmetro. O exemplo a seguir salva a saída em um arquivo local chamado `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
```

O arquivo salvo pode conter linhas em branco. Eles aparecem no final de cada parte do arquivo de log durante o download. Isso geralmente não causa nenhum problema na análise do arquivo de log.

- Para API obter detalhes, consulte [DownloadDbLogFilePortion](#) na Referência de AWS CLI Comandos.

generate-auth-token

O código de exemplo a seguir mostra como usar `generate-auth-token`.

AWS CLI

Para gerar um token de autenticação

O `generate-db-auth-token` exemplo a seguir gera um token de autenticação para uso com a autenticação do IAM banco de dados.

```
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
```

Saída:

```
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
```

- Para API obter detalhes, consulte [GenerateAuthToken](#) na Referência de AWS CLI Comandos.

`generate-db-auth-token`

O código de exemplo a seguir mostra como usar `generate-db-auth-token`.

AWS CLI

Para gerar um token IAM de autenticação

O `generate-db-auth-token` exemplo a seguir gera um token de IAM autenticação para se conectar a um banco de dados.

```
aws rds generate-db-auth-token \  
  --hostname mydb.123456789012.us-east-1.rds.amazonaws.com \  
  --port 3306 \  
  --region us-east-1 \  
  --username db_user
```

Saída:

```
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
```

Para obter mais informações, consulte [Conectando-se à sua instância de banco de dados usando IAM autenticação](#) no Guia RDS do usuário da Amazon e [Conectando-se ao seu cluster de banco de dados usando IAM autenticação](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [GenerateDbAuthToken](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags em um RDS recurso da Amazon

O `list-tags-for-resource` exemplo a seguir lista todas as tags em uma instância de banco de dados.

```
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

Saída:

```
{
  "TagList": [
    {
      "Key": "Environment",
      "Value": "test"
    },
    {
      "Key": "Name",
      "Value": "MyDatabase"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte Como [marcar RDS recursos da Amazon](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

modify-certificates

O código de exemplo a seguir mostra como usar `modify-certificates`.

AWS CLI

Para substituir temporariamente o TLS certificadoSSL/padrão do sistema para novas instâncias de banco de dados

O `modify-certificates` exemplo a seguir substitui temporariamente o TLS certificado SSL / default do sistema para novas instâncias de banco de dados.

```
aws rds modify-certificates \  
--certificate-identifier rds-ca-2019
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Rotação do seu TLS certificadoSSL](#)/no Guia do RDS usuário da Amazon e [Rotação do seu TLS certificadoSSL](#)/no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyCertificates](#) na Referência de AWS CLI Comandos.

modify-current-db-cluster-capacity

O código de exemplo a seguir mostra como usar `modify-current-db-cluster-capacity`.

AWS CLI

Para escalar a capacidade de um cluster de banco de dados Aurora Serverless

O `modify-current-db-cluster-capacity` exemplo a seguir escala a capacidade de um cluster de banco de dados Aurora Serverless para 8.

```
aws rds modify-current-db-cluster-capacity \  
  --db-cluster-identifier mydbcluster \  
  --capacity 8
```

Saída:

```
{  
  "DBClusterIdentifier": "mydbcluster",  
  "PendingCapacity": 8,  
  "CurrentCapacity": 1,  
  "SecondsBeforeTimeout": 300,  
  "TimeoutAction": "ForceApplyCapacityChange"  
}
```

Para obter mais informações, consulte [Dimensionar manualmente a capacidade do cluster de banco de dados Aurora Serverless v1 no Guia do usuário do Amazon Aurora](#).

- Para API obter detalhes, consulte [ModifyCurrentDbClusterCapacity](#) na Referência de AWS CLI Comandos.

modify-db-cluster-endpoint

O código de exemplo a seguir mostra como usar `modify-db-cluster-endpoint`.

AWS CLI

Para modificar um endpoint de cluster de banco de dados personalizado

O `modify-db-cluster-endpoint` exemplo a seguir modifica o endpoint do cluster de banco de dados personalizado especificado.

```
aws rds modify-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --static-members dbinstance1 dbinstance2 dbinstance3
```

Saída:

```
{  
  "DBClusterEndpointIdentifier": "mycustomendpoint",  
  "DBClusterIdentifier": "mydbcluster",  
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",  
  "Endpoint": "mycustomendpoint.cluster-custom-cnpxexample.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"  
}
```

Para obter mais informações, consulte [Amazon Aurora Connection Management](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbClusterEndpoint](#) na Referência de AWS CLI Comandos.

modify-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `modify-db-cluster-parameter-group`.

AWS CLI

Para modificar parâmetros em um grupo de parâmetros de cluster de banco de dados

O `modify-db-cluster-parameter-group` exemplo a seguir modifica os valores dos parâmetros em um grupo de parâmetros do cluster de banco de dados.

```
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"
```

Saída:

```
{  
  "DBClusterParameterGroupName": "mydbclusterpg"  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

modify-db-cluster-snapshot-attribute

O código de exemplo a seguir mostra como usar `modify-db-cluster-snapshot-attribute`.

AWS CLI

Para modificar um atributo de snapshot de cluster de banco de dados

O `modify-db-cluster-snapshot-attribute` exemplo a seguir faz alterações no atributo de snapshot do cluster de banco de dados especificado.

```
aws rds modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier myclustersnapshot \  
  --attribute-name restore \  
  --values-to-add 123456789012
```

Saída:

```
{  
  "DBClusterSnapshotAttributesResult": {
```

```

    "DBClusterSnapshotIdentifier": "myclustersnapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012"
        ]
      }
    ]
  }
}

```

Para obter mais informações, consulte [Restauração a partir de um snapshot de cluster de banco de dados no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbClusterSnapshotAttribute](#) na Referência de AWS CLI Comandos.

modify-db-cluster

O código de exemplo a seguir mostra como usar `modify-db-cluster`.

AWS CLI

Exemplo 1: Para modificar um cluster de banco de dados

O `modify-db-cluster` exemplo a seguir altera a senha do usuário mestre para o cluster de banco de dados chamado `cluster-2` e define o período de retenção de backup para 14 dias. O `--apply-immediately` parâmetro faz com que as alterações sejam feitas imediatamente, em vez de esperar até a próxima janela de manutenção.

```

aws rds modify-db-cluster \
  --db-cluster-identifier cluster-2 \
  --backup-retention-period 14 \
  --master-user-password newpassword99 \
  --apply-immediately

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Modificar um cluster de banco de dados Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Exemplo 2: Para associar um grupo VPC de segurança a um cluster de banco de dados

O `modify-db-instance` exemplo a seguir associa um grupo de VPC segurança específico e remove grupos de segurança de banco de dados de um cluster de banco de dados.

```

aws rds modify-db-cluster \
  --db-cluster-identifier dbName \
  --vpc-security-group-ids sg-ID

```

Saída:

```

{
  "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...
}
}
```

Para obter mais informações, consulte [Controle de acesso com grupos de segurança](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbCluster](#) na Referência de AWS CLI Comandos.

modify-db-instance

O código de exemplo a seguir mostra como usar `modify-db-instance`.

AWS CLI

Exemplo 1: Como modificar uma instância de banco de dados

O `modify-db-instance` exemplo a seguir associa um grupo de opções e um grupo de parâmetros a uma instância de banco de dados Microsoft SQL Server compatível. O parâmetro

`--apply-immediately` faz com que os grupos de opção e parâmetro sejam associados imediatamente em vez de ter que esperar a próxima janela de manutenção.

```
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
```

Saída:

```
{  
  "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
  }
}

```

Para obter mais informações, consulte [Modificar uma RDS instância de banco de dados da Amazon](#) no Guia do RDS usuário da Amazon.

Exemplo 2: Para associar um grupo VPC de segurança a uma instância de banco de dados

O `modify-db-instance` exemplo a seguir associa um grupo de VPC segurança específico e remove grupos de segurança de banco de dados de uma instância de banco de dados:

```

aws rds modify-db-instance \
  --db-instance-identifier dbName \
  --vpc-security-group-ids sg-ID

```

Saída:

```

{
  "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 omitted ...
}
}

```

Para obter mais informações, consulte [Controle de acesso com grupos de segurança](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [ModifyDBInstance](#) na Referência de AWS CLI Comandos.

modify-db-parameter-group

O código de exemplo a seguir mostra como usar `modify-db-parameter-group`.

AWS CLI

Como modificar um grupo de parâmetros de banco de dados

O exemplo de `modify-db-parameter-group` a seguir altera o valor do parâmetro `clr enabled` em um grupo de parâmetros de banco de dados. O parâmetro `--apply-immediately` faz com que o grupo de parâmetros de banco de dados seja modificado imediatamente em vez de ter que esperar a próxima janela de manutenção.

```

aws rds modify-db-parameter-group \
  --db-parameter-group-name test-sqlserver-se-2017 \
  --parameters "ParameterName='clr enabled', ParameterValue=1, ApplyMethod=immediate"

```

Saída:

```
{
```

```
"DBParameterGroupName": "test-sqlserver-se-2017"
}
```

Para obter mais informações, consulte [Modificar parâmetros em um grupo de parâmetros de banco](#) de dados no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [odifyDBParameterGrupo M](#) na Referência de AWS CLI Comandos.

modify-db-proxy-endpoint

O código de exemplo a seguir mostra como usar `modify-db-proxy-endpoint`.

AWS CLI

Para modificar um endpoint de proxy de banco de dados para um RDS banco de dados

O `modify-db-proxy-endpoint` exemplo a seguir modifica um endpoint de proxy de banco de dados `proxyEndpoint` para definir o tempo limite de leitura em 65 segundos.

```
aws rds modify-db-proxy-endpoint \
  --db-proxy-endpoint-name proxyEndpoint \
  --cli-read-timeout 65
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Modificação de um endpoint de proxy](#) no Guia RDS do usuário da Amazon e [Modificação de um endpoint de proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbProxyEndpoint](#) na Referência de AWS CLI Comandos.

modify-db-proxy-target-group

O código de exemplo a seguir mostra como usar `modify-db-proxy-target-group`.

AWS CLI

Para modificar os endpoints de um proxy de banco de dados

O `modify-db-proxy-target-group` exemplo a seguir modifica um grupo-alvo de proxy de banco de dados para definir o máximo de conexões em 80% e o máximo de conexões ociosas em 10%.

```

aws rds modify-db-proxy-target-group \
  --target-group-name default \
  --db-proxy-name proxyExample \
  --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10

```

Saída:

```

{
  "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",
    "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
  }
}

```

Para obter mais informações, consulte [Modificar um RDS proxy](#) no Guia do RDSusuário da Amazon e [Modificar um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbProxyTargetGroup](#) na Referência de AWS CLI Comandos.

modify-db-proxy

O código de exemplo a seguir mostra como usar modify-db-proxy.

AWS CLI

Para modificar um proxy de banco de dados para um RDS banco de dados

O modify-db-proxy exemplo a seguir modifica um proxy de banco de dados nomeado proxyExample SSL para exigir suas conexões.

```

aws rds modify-db-proxy \
  --db-proxy-name proxyExample \
  --require-tls

```

Saída:

```

{
  "DBProxy": {
    "DBProxyName": "proxyExample",
    "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:prx-0123a01b12345c0ab",
    "Status": "modifying"
  }
}

```

```

    "EngineFamily": "PostgreSQL",
    "VpcId": "sg-1234567",
    "VpcSecurityGroupIds": [
      "sg-1234"
    ],
    "VpcSubnetIds": [
      "subnetgroup1",
      "subnetgroup2"
    ],
    "Auth": "[
      {
        "Description": "proxydescription1",
        "AuthScheme": "SECRETS",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
        "IAMAuth": "DISABLED"
      }
    ]",
    "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
    "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
    "RequireTLS": true,
    "IdleClientTimeout": 1800,
    "DebuggingLogging": false,
    "CreateDate": "2023-04-05T16:09:33.452000+00:00",
    "UpdateDate": "2023-04-13T01:49:38.568000+00:00"
  }
}

```

Para obter mais informações, consulte [Modificar um RDS proxy](#) no Guia RDS do usuário da Amazon e [Criar um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbProxy](#) na Referência de AWS CLI Comandos.

modify-db-shard-group

O código de exemplo a seguir mostra como usar `modify-db-shard-group`.

AWS CLI

Exemplo 1: Para modificar um grupo de fragmentos de banco de dados

O `modify-db-shard-group` exemplo a seguir altera a capacidade máxima de um grupo de fragmentos de banco de dados.

```
aws rds modify-db-shard-group \  
  --db-shard-group-identifier my-db-shard-group \  
  --max-acu 1000
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Amazon Aurora DB Clusters](#) no Guia do usuário do Amazon Aurora.

Exemplo 2: Para descrever seus grupos de fragmentos de banco de dados

O `describe-db-shard-groups` exemplo a seguir recupera os detalhes dos seus grupos de fragmentos de banco de dados depois que você executa o `modify-db-shard-group` comando. A capacidade máxima do grupo de fragmentos de banco de dados agora `my-db-shard-group` é de 1000 unidades ACUs de capacidade Aurora ().

```
aws rds describe-db-shard-groups
```

Saída:

```
{  
  "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"
  }
]
}

```

Para obter mais informações, consulte [Amazon Aurora DB Clusters](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyDbShardGroup](#) na Referência de AWS CLI Comandos.

modify-db-snapshot-attribute

O código de exemplo a seguir mostra como usar `modify-db-snapshot-attribute`.

AWS CLI

Exemplo 1: Para habilitar duas AWS contas para restaurar um DB snapshot

O `modify-db-snapshot-attribute` exemplo a seguir concede permissão a duas AWS contas, com os identificadores 111122223333 e 444455556666, para restaurar o DB snapshot chamado. `mydbsnapshot`

```

aws rds modify-db-snapshot-attribute \
  --db-snapshot-identifier mydbsnapshot \
  --attribute-name restore \

```

```
--values-to-add {"111122223333","444455556666"}
```

Saída:

```
{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "111122223333",
          "444455556666"
        ]
      }
    ]
  }
}
```

Para obter mais informações, consulte [Compartilhamento de um snapshot](#) no Guia do RDS usuário da Amazon.

Exemplo 2: Para evitar que uma AWS conta restaure um DB snapshot

O `modify-db-snapshot-attribute` exemplo a seguir remove a permissão de uma AWS conta específica para restaurar o DB snapshot chamado `mydbsnapshot`. Ao especificar uma única conta, o identificador da conta não pode ser delimitado por aspas ou chaves.

```
aws rds modify-db-snapshot-attribute \
  --db-snapshot-identifier mydbsnapshot \
  --attribute-name restore \
  --values-to-remove 444455556666
```

Saída:

```
{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
```

```

        "111122223333"
      ]
    }
  ]
}

```

Para obter mais informações, consulte [Compartilhamento de um snapshot](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [ModifyDbSnapshotAttribute](#) na Referência de AWS CLI Comandos.

modify-db-snapshot-attributes

O código de exemplo a seguir mostra como usar `modify-db-snapshot-attributes`.

AWS CLI

Para modificar um atributo de DB snapshot

O `modify-db-snapshot-attribute` exemplo a seguir permite dois identificadores de AWS conta 111122223333 e 444455556666, restaurar o DB snapshot chamado. `mydbsnapshot`

```

aws rds modify-db-snapshot-attribute \
  --db-snapshot-identifier mydbsnapshot \
  --attribute-name restore \
  --values-to-add ['"111122223333", "444455556666"]'

```

Saída:

```

{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "111122223333",
          "444455556666"
        ]
      }
    ]
  }
}

```

```
    ]  
  }  
}
```

Para obter mais informações, consulte [Compartilhamento de um snapshot](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [ModifyDbSnapshotAttributes](#) na Referência de AWS CLI Comandos.

modify-db-snapshot

O código de exemplo a seguir mostra como usar `modify-db-snapshot`.

AWS CLI

Para modificar um DB snapshot

O `modify-db-snapshot` exemplo a seguir atualiza um snapshot do Postge SQL 10.6 chamado `db5-snapshot-upg-test` Postgre 11.7. SQL A nova versão do mecanismo de banco de dados é exibida após a conclusão da atualização do snapshot e seu status está disponível.

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifier db5-snapshot-upg-test \  
  --engine-version 11.7
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Atualizando um SQL DB snapshot do Postgre](#) no Guia do usuário da Amazon RDS.

- Para API obter detalhes, consulte [ModifyDbSnapshot](#) na Referência de AWS CLI Comandos.

modify-db-subnet-group

O código de exemplo a seguir mostra como usar `modify-db-subnet-group`.

AWS CLI

Para modificar um grupo de sub-redes de banco de dados

O `modify-db-subnet-group` exemplo a seguir adiciona uma sub-rede com o ID `subnet-08e41f9e230222222` ao grupo de sub-redes de banco de dados chamado `mysubnetgroup`. Para manter as sub-redes existentes no grupo de sub-redes, inclua-as IDs como valores na opção. `--subnet-ids`. Certifique-se de ter sub-redes com pelo menos duas zonas de disponibilidade diferentes no grupo de sub-redes de banco de dados.

```

aws rds modify-db-subnet-group \
  --db-subnet-group-name mysubnetgroup \
  --subnet-ids
  '["subnet-0a1dc4e1a6f123456", "subnet-070dd7ecb3aaaaaaa", "subnet-00f5b198bc0abcdef", "subnet-

```

Saída:

```

{
  "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-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-2:534026745191:subgrp:mysubnetgroup"
}
```

Para obter mais informações, consulte [Etapa 3: Criar um grupo de sub-rede de banco](#) de dados no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [ModifyDbSubnetGroup](#) na Referência de AWS CLI Comandos.

modify-event-subscription

O código de exemplo a seguir mostra como usar `modify-event-subscription`.

AWS CLI

Para modificar a assinatura de um evento

O `modify-event-subscription` exemplo a seguir desativa a assinatura do evento especificado, de forma que não publique mais notificações para o tópico especificado do Amazon Simple Notification Service.

```
aws rds modify-event-subscription \  
  --subscription-name my-instance-events \  
  --no-enabled
```

Saída:

```
{  
  "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  
  }  
}
```

- Para API obter detalhes, consulte [ModifyEventSubscription](#) na Referência de AWS CLI Comandos.

modify-global-cluster

O código de exemplo a seguir mostra como usar `modify-global-cluster`.

AWS CLI

Para modificar um cluster de banco de dados global

O `modify-global-cluster` exemplo a seguir ativa a proteção contra exclusão de um cluster de banco de dados global compatível com o Aurora SQL My.

```
aws rds modify-global-cluster \  
  --global-cluster-identifier myglobalcluster \  
  --deletion-protection
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Gerenciamento de um banco de dados global do Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ModifyGlobalCluster](#) na Referência de AWS CLI Comandos.

promote-read-replica-db-cluster

O código de exemplo a seguir mostra como usar `promote-read-replica-db-cluster`.

AWS CLI

Para promover um cluster de banco de dados, leia a réplica

O `promote-read-replica-db-cluster` exemplo a seguir promove a réplica de leitura especificada para se tornar um cluster de banco de dados independente.

```
aws rds promote-read-replica-db-cluster \  
--db-cluster-identifier mydbcluster-1
```

Saída:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1c"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "mydbcluster-1",  
    "...some output truncated..."  
  }  
}
```

Para obter mais informações, consulte [Promoção de uma réplica de leitura para ser um cluster de banco](#) de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [PromoteReadReplicaDbCluster](#) na Referência de AWS CLI Comandos.

promote-read-replica

O código de exemplo a seguir mostra como usar `promote-read-replica`.

AWS CLI

Para promover uma réplica de leitura

O `promote-read-replica` exemplo a seguir promove a réplica de leitura especificada para se tornar uma instância de banco de dados independente.

```
aws rds promote-read-replica \  
--db-instance-identifier test-instance-repl
```

Saída:

```
{
  "DBInstance": {
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
    "StorageType": "standard",
    "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
    "DBInstanceStatus": "modifying",
    ...some output truncated...
  }
}
```

- Para API obter detalhes, consulte [PromoteReadReplica](#) na Referência de AWS CLI Comandos.

purchase-reserved-db-instance

O código de exemplo a seguir mostra como usar `purchase-reserved-db-instance`.

AWS CLI

Para comprar uma oferta de instância de banco de dados reservada

O `purchase-reserved-db-instances-offering` exemplo a seguir compra uma oferta de instância de banco de dados reservada. `reserved-db-instances-offering-id` Deve ser um ID de oferta válido, conforme retornado pelo `describe-reserved-db-instances-offering` comando.

```
aws rds purchase-reserved-db-instances -offering -- reserved-db-instances-offering id
438012d3-4a52-4cc7-b2e3-8dff72e0e706
```

- Para API obter detalhes, consulte [PurchaseReservedDbInstance](#) na Referência de AWS CLI Comandos.

purchase-reserved-db-instances-offerings

O código de exemplo a seguir mostra como usar `purchase-reserved-db-instances-offerings`.

AWS CLI

Exemplo 1: encontrar uma instância de banco de dados reservada para comprar

O `describe-reserved-db-instances-offerings` exemplo a seguir lista as instâncias reservadas do My SQL DB disponíveis com a classe de instância `db.t2.micro` e uma duração de um ano. O ID da oferta é necessário para comprar uma instância de banco de dados reservada.

```
aws rds describe-reserved-db-instances-offerings \  
  --product-description mysql \  
  --db-instance-class db.t2.micro \  
  --duration 1
```

Saída:

```
{  
  "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 ...  
  ]  
}
```

Para obter mais informações, consulte [Instâncias de banco de dados reservadas para a Amazon RDS](#) no Guia RDS do usuário da Amazon.

Exemplo 2: Para comprar uma instância de banco de dados reservada

O `purchase-reserved-db-instances-offering` exemplo a seguir mostra como comprar a oferta de instância de banco de dados reservada do exemplo anterior.

```
aws rds purchase-reserved-db-instances --offering --reserved-db-instances-offering id 8ba30be1-b9ec-447f-8f23-6114e3f4c7b4
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Instâncias de banco de dados reservadas para a Amazon RDS](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [PurchaseReservedDbInstancesOfferings](#) na Referência de AWS CLI Comandos.

reboot-db-instance

O código de exemplo a seguir mostra como usar `reboot-db-instance`.

AWS CLI

Como reinicializar uma instância de banco de dados

O exemplo de `reboot-db-instance` a seguir inicia uma reinicialização da instância de banco de dados especificada.

```
aws rds reboot-db-instance \  
  --db-instance-identifier test-mysql-instance
```

Saída:

```
{  
  "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...  
  }  
}
```

Para obter mais informações, consulte [Reinicializando uma instância](#) de banco de dados no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [RebootDBInstance](#) na Referência de AWS CLI Comandos.

reboot-db-shard-group

O código de exemplo a seguir mostra como usar `reboot-db-shard-group`.

AWS CLI

Exemplo 1: Para reinicializar um grupo de fragmentos de banco de dados

O `reboot-db-shard-group` exemplo a seguir reinicializa um grupo de fragmentos de banco de dados.

```
aws rds reboot-db-shard-group \  
  --db-shard-group-identifier my-db-shard-group
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Reinicializar um cluster de banco de dados Amazon Aurora ou uma instância de banco de dados Amazon Aurora no Guia do usuário do Amazon Aurora](#).

Exemplo 2: Para descrever seus grupos de fragmentos de banco de dados

O `describe-db-shard-groups` exemplo a seguir recupera os detalhes dos seus grupos de fragmentos de banco de dados depois que você executa o `reboot-db-shard-group` comando. O grupo de fragmentos de banco de dados agora `my-db-shard-group` está sendo reinicializado.

```
aws rds describe-db-shard-groups
```

Saída:

```
{  
  "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"
    }
  ]
}

```

Para obter mais informações, consulte [Reinicializar um cluster de banco de dados Amazon Aurora ou uma instância de banco de dados Amazon Aurora no Guia do usuário do Amazon Aurora](#).

- Para API obter detalhes, consulte [RebootDbShardGroup](#) na Referência de AWS CLI Comandos.

register-db-proxy-targets

O código de exemplo a seguir mostra como usar `register-db-proxy-targets`.

AWS CLI

Para registrar um proxy de banco de dados com um banco de dados

O `register-db-proxy-targets` exemplo a seguir cria a associação entre um banco de dados e um proxy.

```
aws rds register-db-proxy-targets \
```

```
--db-proxy-name proxyExample \
--db-cluster-identifiers database-5
```

Saída:

```
{
  "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"
      }
    }
  ]
}
```

Para obter mais informações, consulte [Criação de um RDS proxy](#) no Guia do RDS usuário da Amazon e [Criação de um RDS proxy](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RegisterDbProxyTargets](#) na Referência de AWS CLI Comandos.

remove-from-global-cluster

O código de exemplo a seguir mostra como usar `remove-from-global-cluster`.

AWS CLI

Para separar um cluster secundário do Aurora de um cluster de banco de dados global do Aurora

O `remove-from-global-cluster` exemplo a seguir separa um cluster secundário do Aurora de um cluster de banco de dados global do Aurora. O cluster deixa de ser somente leitura e passa a ser um cluster autônomo com capacidade de leitura e gravação.

```
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
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Removendo um cluster de um banco de dados global do Amazon Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RemoveFromGlobalCluster](#) na Referência de AWS CLI Comandos.

remove-option-from-option-group

O código de exemplo a seguir mostra como usar `remove-option-from-option-group`.

AWS CLI

Para excluir uma opção de um grupo de opções

O `remove-option-from-option-group` exemplo a seguir remove a OEM opção `demyoptiongroup`.

```
aws rds remove-option-from-option-group \
  --option-group-name myoptiongroup \
  --options OEM \
  --apply-immediately
```

Saída:

```
{
  "OptionGroup": {
    "OptionGroupName": "myoptiongroup",
    "OptionGroupDescription": "Test",
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "19",
    "Options": [],
    "AllowsVpcAndNonVpcInstanceMemberships": true,
    "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
  }
}
```

Para obter mais informações, consulte [Removendo uma opção de um grupo de opções](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RemoveOptionFromOptionGroup](#) na Referência de AWS CLI Comandos.

remove-role-from-db-cluster

O código de exemplo a seguir mostra como usar `remove-role-from-db-cluster`.

AWS CLI

Para desassociar uma função de AWS Identity and Access Management (IAM) de um cluster de banco de dados

O `remove-role-from-db-cluster` exemplo a seguir remove uma função de um cluster de banco de dados.

```
aws rds remove-role-from-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Este comando não produz saída.

Para obter mais informações, consulte [Associar uma IAM função a um cluster Amazon Aurora SQL My DB](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RemoveRoleFromDbCluster](#) na Referência de AWS CLI Comandos.

remove-role-from-db-instance

O código de exemplo a seguir mostra como usar `remove-role-from-db-instance`.

AWS CLI

Para desassociar uma função de AWS Identity and Access Management (IAM) de uma instância de banco de dados

O `remove-role-from-db-instance` exemplo a seguir remove a função nomeada `rds-s3-integration-role` de uma instância de banco de dados Oracle chamada `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Desabilitar a integração RDS SQL do servidor com o S3](#) no Guia RDS do usuário da Amazon.

- Para API obter detalhes, consulte [RemoveRoleFromDbInstance](#) na Referência de AWS CLI Comandos.

remove-source-identifier-from-subscription

O código de exemplo a seguir mostra como usar `remove-source-identifier-from-subscription`.

AWS CLI

Para remover um identificador de origem de uma assinatura

O `remove-source-identifier` exemplo a seguir remove o identificador de origem especificado de uma assinatura existente.

```
aws rds remove-source-identifier-from-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

Saída:

```
{  
  "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
```

```
}  
}
```

- Para API obter detalhes, consulte [RemoveSourceIdentifierFromSubscription](#) na Referência de AWS CLI Comandos.

remove-tags-from-resource

O código de exemplo a seguir mostra como usar `remove-tags-from-resource`.

AWS CLI

Como remover as tags de um recurso

O `remove-tags-from-resource` exemplo a seguir remove as tags de um recurso.

```
aws rds remove-tags-from-resource \  
  --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \  
  --tag-keys Name Environment
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar RDS recursos da Amazon](#) no Guia do RDSusuário da Amazon e [Marcar RDS recursos da Amazon](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RemoveTagsFromResource](#) na Referência de AWS CLI Comandos.

reset-db-cluster-parameter-group

O código de exemplo a seguir mostra como usar `reset-db-cluster-parameter-group`.

AWS CLI

Exemplo 1: Para redefinir todos os parâmetros para seus valores padrão

O `reset-db-cluster-parameter-group` exemplo a seguir redefine todos os valores de parâmetros em um grupo de parâmetros de cluster de banco de dados criado pelo cliente para seus valores padrão.

```
aws rds reset-db-cluster-parameter-group \  
  --parameter-group-name mydbparametergroup
```



```
--db-cluster-parameter-group-name mydbclpg \  
--reset-all-parameters
```

Saída:

```
{  
  "DBClusterParameterGroupName": "mydbclpg"  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

Exemplo 2: Para redefinir parâmetros específicos para seus valores padrão

O `reset-db-cluster-parameter-group` exemplo a seguir redefine os valores de parâmetros específicos para seus valores padrão em um grupo de parâmetros de cluster de banco de dados criado pelo 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"
```

Saída:

```
{  
  "DBClusterParameterGroupName": "mydbclpg"  
}
```

Para obter mais informações, consulte Como [trabalhar com grupos de parâmetros de banco de dados e grupos de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ResetDbClusterParameterGroup](#) na Referência de AWS CLI Comandos.

reset-db-parameter-group

O código de exemplo a seguir mostra como usar `reset-db-parameter-group`.

AWS CLI

Exemplo 1: Para redefinir todos os parâmetros para seus valores padrão

O `reset-db-parameter-group` exemplo a seguir redefine todos os valores de parâmetros em um grupo de parâmetros de banco de dados criado pelo cliente para seus valores padrão.

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --reset-all-parameters
```

Saída:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

Para obter mais informações, consulte [Trabalhando com grupos de parâmetros de banco de dados](#) no Guia do RDS usuário da Amazon e [Trabalhando com grupos de parâmetros de banco de dados e grupos de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

Exemplo 2: Para redefinir parâmetros específicos para seus valores padrão

O `reset-db-parameter-group` exemplo a seguir redefine os valores de parâmetros específicos para seus valores padrão em um grupo de parâmetros de banco de dados criado pelo cliente.

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
               "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Saída:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

Para obter mais informações, consulte [Trabalhando com grupos de parâmetros de banco de dados](#) no Guia do RDS usuário da Amazon e [Trabalhando com grupos de parâmetros de banco de dados e grupos de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

[de dados e grupos de parâmetros de cluster](#) de banco de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [ResetDbParameterGroup](#) na Referência de AWS CLI Comandos.

restore-db-cluster-from-s3

O código de exemplo a seguir mostra como usar `restore-db-cluster-from-s3`.

AWS CLI

Para restaurar um cluster de banco de dados Amazon Aurora a partir do Amazon S3

O `restore-db-cluster-from-s3` exemplo a seguir restaura um cluster de banco de dados compatível com o Amazon Aurora SQL My versão 5.7 a partir de um arquivo de backup do SQL My 5.7 DB no 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
```

Saída:

```
{  
  "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": []
  }
}

```

Para obter mais informações, consulte [Migração de dados do My SQL usando um bucket do Amazon S3](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RestoreDbClusterFromS3](#) na Referência de AWS CLI Comandos.

restore-db-cluster-from-snapshot

O código de exemplo a seguir mostra como usar `restore-db-cluster-from-snapshot`.

AWS CLI

Para restaurar um cluster de banco de dados a partir de um snapshot

O seguinte `restore-db-cluster-from-snapshot` restaura um cluster de banco de dados Aurora SQL Postgre compatível com o SQL Postgre versão 10.7 a partir de um snapshot de cluster de banco de dados chamado. `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
```

Saída:

```
{  
  "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": "Z1PVIIF0EXAMPLE",
    "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": []
  }
}

```

Para obter mais informações, consulte [Restauração a partir de um snapshot de cluster de banco de dados no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RestoreDbClusterFromSnapshot](#) na Referência de AWS CLI Comandos.

restore-db-cluster-to-point-in-time

O código de exemplo a seguir mostra como usar `restore-db-cluster-to-point-in-time`.

AWS CLI

Para restaurar um cluster de banco de dados em um horário especificado

O `restore-db-cluster-to-point-in-time` exemplo a seguir restaura o cluster de banco de dados nomeado `database-4` para a última hora possível. Usar `copy-on-write` como tipo

de restauração restaura o novo cluster de banco de dados como um clone do cluster de banco de dados de origem.

```
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
```

Saída:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "sample-cluster-clone",  
    "DBClusterParameterGroup": "default.aurora-postgresql10",  
    "DBSubnetGroup": "default",  
    "Status": "creating",  
    "Endpoint": "sample-cluster-clone.cluster-#####.us-  
west-2.rds.amazonaws.com",  
    "ReaderEndpoint": "sample-cluster-clone.cluster-ro-#####.us-  
west-2.rds.amazonaws.com",  
    "MultiAZ": false,  
    "Engine": "aurora-postgresql",  
    "EngineVersion": "10.7",  
    "Port": 5432,  
    "MasterUsername": "postgres",  
    "PreferredBackupWindow": "09:33-10:03",  
    "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",  
    "ReadReplicaIdentifiers": [],  
    "DBClusterMembers": [],  
    "VpcSecurityGroups": [  
      {  
        "VpcSecurityGroupId": "sg-#####",  
        "Status": "active"  
      }  
    ]  
  }  
}
```

```

    ],
    "HostedZoneId": "Z1PVIIF0EXAMPLE",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
    "DbClusterResourceId": "cluster-BIZ77GDSA2XBSTNPFW1EXAMPLE",
    "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
clone",
    "AssociatedRoles": [],
    "IAMDatabaseAuthenticationEnabled": false,
    "CloneGroupId": "8d19331a-099a-45a4-b4aa-11aa22bb33cc44dd",
    "ClusterCreateTime": "2020-03-10T19:57:38.967Z",
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "CopyTagsToSnapshot": false,
    "CrossAccountClone": false
  }
}

```

Para obter mais informações, consulte [Restauração de um cluster de banco de dados em um horário especificado no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [RestoreDbClusterToPointInTime](#) na Referência de AWS CLI Comandos.

restore-db-instance-from-db-snapshot

O código de exemplo a seguir mostra como usar `restore-db-instance-from-db-snapshot`.

AWS CLI

Para restaurar uma instância de banco de dados a partir de um DB snapshot

O `restore-db-instance-from-db-snapshot` exemplo a seguir cria uma nova instância de banco de dados nomeada `db7-new-instance` com a classe de instância de `db.t3.small` banco de dados do DB snapshot especificado. A instância de banco de dados de origem da qual o snapshot foi obtido usa uma classe de instância de banco de dados obsoleta, então você não pode atualizá-la.

```

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
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Restauração a partir de um DB snapshot no Guia RDS](#) do usuário da Amazon.

- Para API obter detalhes, consulte [RestoreDbInstanceFromDbSnapshot](#) na Referência de AWS CLI Comandos.

restore-db-instance-from-s3

O código de exemplo a seguir mostra como usar `restore-db-instance-from-s3`.

AWS CLI

Para restaurar uma instância de banco de dados a partir de um backup no Amazon S3

O `restore-db-instance-from-s3` exemplo a seguir cria uma nova instância de banco de dados nomeada `restored-test-instance` a partir de um backup existente no bucket do `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
```

- Para API obter detalhes, consulte [RestoreDbInstanceFromS3](#) na Referência de AWS CLI Comandos.

`restore-db-instance-to-point-in-time`

O código de exemplo a seguir mostra como usar `restore-db-instance-to-point-in-time`.

AWS CLI

Exemplo 1: Para restaurar uma instância de banco de dados em um momento específico

O `restore-db-instance-to-point-in-time` exemplo a seguir é restaurado `test-instance` em uma nova instância de banco de dados chamada `restored-test-instance`, a partir do horário especificado.

```
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
```

Saída:

```
{  
  "DBInstance": {  
    "AllocatedStorage": 20,
```

```

    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-
instance",
    "DBInstanceStatus": "creating",
    "DBInstanceIdentifier": "restored-test-instance",
    ...some output omitted...
  }
}

```

Para obter mais informações, consulte [Restauração de uma instância de banco de dados em um horário especificado](#) no Guia do RDS usuário da Amazon.

Exemplo 2: Para restaurar uma instância de banco de dados em um horário especificado a partir de um backup replicado

O `restore-db-instance-to-point-in-time` exemplo a seguir restaura uma instância de banco de dados Oracle para o horário especificado a partir de um backup automatizado replicado.

```

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

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Restauração em um horário especificado a partir de um backup replicado no Guia RDS](#) do usuário da Amazon.

- Para API obter detalhes, consulte [RestoreDbInstanceToPointInTime](#) na Referência de AWS CLI Comandos.

start-activity-stream

O código de exemplo a seguir mostra como usar `start-activity-stream`.

AWS CLI

Para iniciar um fluxo de atividades do banco de dados

O `start-activity-stream` exemplo a seguir inicia um fluxo de atividades assíncrono para monitorar um cluster do Aurora chamado. `my-pg-cluster`

```

aws rds start-activity-stream \
  --region us-east-1 \
  --mode async \
  --kms-key-id arn:aws:kms:us-east-1:123456789012:key/a12c345d-6ef7-890g-
h123-456i789jk011 \
  --resource-arn arn:aws:rds:us-east-1:123456789012:cluster:my-pg-cluster \
  --apply-immediately

```

Saída:

```

{
  "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/a12c345d-6ef7-890g-
h123-456i789jk011",
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2NOPQ3R4S",
  "Status": "starting",
}

```

```
"Mode": "async",
"ApplyImmediately": true
}
```

Para obter mais informações, consulte [Iniciando um fluxo de atividades do banco](#) de dados no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [StartActivityStream](#) na Referência de AWS CLI Comandos.

start-db-cluster

O código de exemplo a seguir mostra como usar `start-db-cluster`.

AWS CLI

Para iniciar um cluster de banco de dados

O `start-db-cluster` exemplo a seguir inicia um cluster de banco de dados e suas instâncias de banco de dados.

```
aws rds start-db-cluster \
  --db-cluster-identifier mydbcluster
```

Saída:

```
{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1e",
      "us-east-1b"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydb",
    "DBClusterIdentifier": "mydbcluster",
    ...some output truncated...
  }
}
```

Para obter mais informações, consulte Como [interromper e iniciar um cluster de banco de dados Amazon Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [StartDbCluster](#) na Referência de AWS CLI Comandos.

start-db-instance-automated-backups-replication

O código de exemplo a seguir mostra como usar `start-db-instance-automated-backups-replication`.

AWS CLI

Para habilitar backups automatizados entre regiões

O `start-db-instance-automated-backups-replication` exemplo a seguir replica backups automatizados de uma instância de banco de dados na região Leste dos EUA (Norte da Virgínia) para Oeste dos EUA (Oregon). O período de retenção do backup é de 14 dias.

```
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
```

Saída:

```
{  
  "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-jkib2gfq5rv7replzadausbrktni2bn4example"
  }
}
```

Para obter mais informações, consulte [Habilitar backups automatizados entre regiões](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [StartDbInstanceAutomatedBackupsReplication](#) na Referência de AWS CLI Comandos.

start-db-instance

O código de exemplo a seguir mostra como usar `start-db-instance`.

AWS CLI

Para iniciar uma instância de banco de dados

O `start-db-instance` exemplo a seguir inicia a instância de banco de dados especificada.

```
aws rds start-db-instance \
  --db-instance-identifier test-instance
```

Saída:

```
{
  "DBInstance": {
    "DBInstanceStatus": "starting",
    ...some output truncated...
  }
}
```

- Para API obter detalhes, consulte [StartDbInstance](#) na Referência de AWS CLI Comandos.

start-export-task

O código de exemplo a seguir mostra como usar `start-export-task`.

AWS CLI

Para exportar um snapshot para o Amazon S3

O `start-export-task` exemplo a seguir exporta um DB snapshot chamado `db5-snapshot-test` para o bucket do Amazon S3 chamado `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
```

Saída:

```
{  
  "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  
}
```

Para obter mais informações, consulte [Exportação de um snapshot para um bucket do Amazon S3](#) no Guia do usuário da Amazon RDS.

- Para API obter detalhes, consulte [StartExportTask](#) na Referência de AWS CLI Comandos.

stop-activity-stream

O código de exemplo a seguir mostra como usar `stop-activity-stream`.

AWS CLI

Para interromper um fluxo de atividades do banco de dados

O `stop-activity-stream` exemplo a seguir interrompe um stream de atividades em um cluster do Aurora chamado `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
```

Saída:

```
{  
  "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-  
h123-456i789jk011",  
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH1JKLM2NOPQ3R4S",  
  "Status": "stopping"  
}
```

Para obter mais informações, consulte [Interromper um fluxo de atividades](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [StopActivityStream](#) na Referência de AWS CLI Comandos.

stop-db-cluster

O código de exemplo a seguir mostra como usar `stop-db-cluster`.

AWS CLI

Para parar um cluster de banco de dados

O `stop-db-cluster` exemplo a seguir interrompe um cluster de banco de dados e suas instâncias de banco de dados.

```
aws rds stop-db-cluster \  
  --db-cluster-identifier mydbcluster
```

Saída:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1e",  
      "us-east-1b"  
    ]  
  }  
}
```

```

    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydb",
    "DBClusterIdentifier": "mydbcluster",
    ...some output truncated...
  }
}

```

Para obter mais informações, consulte Como [interromper e iniciar um cluster de banco de dados Amazon Aurora no Guia](#) do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [StopDbCluster](#) na Referência de AWS CLI Comandos.

stop-db-instance-automated-backups-replication

O código de exemplo a seguir mostra como usar `stop-db-instance-automated-backups-replication`.

AWS CLI

Para parar de replicar backups automatizados

O seguinte `stop-db-instance-automated-backups-replication` encerra a replicação de backups automatizados para a região Oeste dos EUA (Oregon). Os backups replicados são retidos de acordo com o período de retenção de backup definido.

```

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"

```

Saída:

```

{
  "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-jkib2gfq5rv7replzadausbrktni2bn4example"
  }
}

```

Para obter mais informações, consulte Como [interromper a replicação automática de backup](#) no Guia do RDS usuário da Amazon.

- Para API obter detalhes, consulte [StopDbInstanceAutomatedBackupsReplication](#) na Referência de AWS CLI Comandos.

stop-db-instance

O código de exemplo a seguir mostra como usar stop-db-instance.

AWS CLI

Para parar uma instância de banco de dados

O stop-db-instance exemplo a seguir interrompe a instância de banco de dados especificada.

```

aws rds stop-db-instance \
  --db-instance-identifier test-instance

```

Saída:

```

{
  "DBInstance": {
    "DBInstanceStatus": "stopping",

```

```

    ...some output truncated...
  }
}

```

- Para API obter detalhes, consulte [StopDbInstance](#) na Referência de AWS CLI Comandos.

switchover-blue-green-deployment

O código de exemplo a seguir mostra como usar `switchover-blue-green-deployment`.

AWS CLI

Exemplo 1: Para alternar uma implantação azul/verde para uma RDS instância de banco de dados

O `switchover-blue-green-deployment` exemplo a seguir promove o ambiente verde especificado como o novo ambiente de produção.

```

aws rds switchover-blue-green-deployment \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
  --switchover-timeout 300

```

Saída:

```

{
  "BlueGreenDeployment": {
    "BlueGreenDeploymentIdentifier": "bgd-v53303651eexfake",
    "BlueGreenDeploymentName": "bgd-cli-test-instance",
    "Source": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "Target": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-blhile",
    "SwitchoverDetails": [
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-green-blhile",
        "Status": "AVAILABLE"
      },
      {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance-replica-1",

```

```
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-1-green-k5fv7u",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-2-green-ggsh8m",
        "Status": "AVAILABLE"
    },
    {
        "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3",
        "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-db-
instance-replica-3-green-o2vwm0",
        "Status": "AVAILABLE"
    }
],
"Tasks": [
    {
        "Name": "CREATING_READ_REPLICA_OF_SOURCE",
        "Status": "COMPLETED"
    },
    {
        "Name": "DB_ENGINE_VERSION_UPGRADE",
        "Status": "COMPLETED"
    },
    {
        "Name": "CONFIGURE_BACKUPS",
        "Status": "COMPLETED"
    },
    {
        "Name": "CREATING_TOPOLOGY_OF_SOURCE",
        "Status": "COMPLETED"
    }
],
"Status": "SWITCHOVER_IN_PROGRESS",
"CreateTime": "2022-02-25T22:33:22.225000+00:00"
}
}
```

Para obter mais informações, consulte [Alternando uma implantação azul/verde no Guia RDS](#) do usuário da Amazon.

Exemplo 2: Para promover uma implantação azul/verde para um cluster Aurora My DB SQL

O `switchover-blue-green-deployment` exemplo a seguir promove o ambiente verde especificado como o novo ambiente de produção.

```
aws rds switchover-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \  
  --switchover-timeout 300
```

Saída:

```
{  
  "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-np1ikl",
      "Status": "AVAILABLE"
    },
    {
      "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
      "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-miszlf",
      "Status": "AVAILABLE"
    }
  ],
  "Tasks": [
    {
      "Name": "CREATING_READ_REPLICA_OF_SOURCE",
      "Status": "COMPLETED"
    },
    {
      "Name": "DB_ENGINE_VERSION_UPGRADE",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
      "Status": "COMPLETED"
    },
    {
      "Name": "CREATE_CUSTOM_ENDPOINTS",
      "Status": "COMPLETED"
    }
  ],
  "Status": "SWITCHOVER_IN_PROGRESS",
  "CreateTime": "2022-02-25T22:38:49.522000+00:00"
}
```

```
}
```

Para obter mais informações, consulte [Alternando uma implantação azul/verde](#) no Guia do usuário do Amazon Aurora.

- Para API obter detalhes, consulte [SwitchoverBlueGreenDeployment](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon RDS Data Service usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon RDS Data Service.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-execute-statement

O código de exemplo a seguir mostra como usar `batch-execute-statement`.

AWS CLI

Para executar uma SQL instrução em lote

O `batch-execute-statement` exemplo a seguir executa uma SQL instrução em lote em uma matriz de dados com um conjunto de parâmetros.

```
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" \  
  --statement "SELECT * FROM mytable" \  
  --parameters "['myparam']" \  
  --output-mode 'stream' \  
  --output-file 'output.txt' \  
  --output-format 'text' \  
  --output-encoding 'utf-8' \  
  --output-headers 'true' \  
  --output-truncate 'true' \  
  --output-truncate-length 1000 \  
  --output-truncate-words 1000 \  
  --output-truncate-rows 1000 \  
  --output-truncate-columns 1000 \  
  --output-truncate-headers 1000 \  
  --output-truncate-words 1000 \  
  --output-truncate-rows 1000 \  
  --output-truncate-columns 1000
```



```
--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\"}}]]]"
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando os dados API do Aurora Serverless no Guia do usuário](#) da Amazon. RDS

- Para API obter detalhes, consulte [BatchExecuteStatement](#) na Referência de AWS CLI Comandos.

begin-transaction

O código de exemplo a seguir mostra como usar begin-transaction.

AWS CLI

Para iniciar uma SQL transação

O begin-transaction exemplo a seguir inicia uma SQL transação.

```
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"
```

Saída:

```
{
  "transactionId": "ABC1234567890xyz"
}
```

Para obter mais informações, consulte [Usando os dados API do Aurora Serverless no Guia do usuário](#) da Amazon. RDS

- Para API obter detalhes, consulte [BeginTransaction](#) na Referência de AWS CLI Comandos.

commit-transaction

O código de exemplo a seguir mostra como usar `commit-transaction`.

AWS CLI

Para confirmar uma SQL transação

O `commit-transaction` exemplo a seguir encerra a SQL transação especificada e confirma as alterações feitas como parte dela.

```
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"
```

Saída:

```
{  
  "transactionStatus": "Transaction Committed"  
}
```

Para obter mais informações, consulte [Usando os dados API do Aurora Serverless no Guia do usuário](#) da Amazon. RDS

- Para API obter detalhes, consulte [CommitTransaction](#) na Referência de AWS CLI Comandos.

execute-statement

O código de exemplo a seguir mostra como usar `execute-statement`.

AWS CLI

Exemplo 1: Para executar uma SQL declaração que faz parte de uma transação

O `execute-statement` exemplo a seguir executa uma SQL declaração que faz parte de uma transação.

```
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" \  
  --statement "SELECT * FROM mytable"
```

```
--sql "update mytable set quantity=5 where id=201" \
--transaction-id "ABC1234567890xyz"
```

Saída:

```
{
  "numberOfRecordsUpdated": 1
}
```

Exemplo 2: Para executar uma SQL instrução com parâmetros

O execute-statement exemplo a seguir executa uma SQL instrução com parâmetros.

```
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\"}}]"
```

Saída:

```
{
  "numberOfRecordsUpdated": 1
}
```

Para obter mais informações, consulte [Usando os dados API do Aurora Serverless no Guia do usuário](#) da Amazon. RDS

- Para API obter detalhes, consulte [ExecuteStatement](#) na Referência de AWS CLI Comandos.

rollback-transaction

O código de exemplo a seguir mostra como usar rollback-transaction.

AWS CLI

Para reverter uma SQL transação

O rollback-transaction exemplo a seguir reverte a SQL transação especificada.

```
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"
```

Saída:

```
{  
  "transactionStatus": "Rollback Complete"  
}
```

Para obter mais informações, consulte [Usando os dados API do Aurora Serverless no Guia do usuário](#) da Amazon. RDS

- Para API obter detalhes, consulte [RollbackTransaction](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon RDS Performance Insights usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon RDS Performance Insights.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-dimension-keys

O código de exemplo a seguir mostra como usar `describe-dimension-keys`.

AWS CLI

Para descrever as chaves de dimensão

Este exemplo solicita os nomes de todos os eventos de espera. Os dados são resumidos pelo nome do evento e pelos valores agregados desses eventos durante o período especificado.

Comando:

```
aws pi describe-dimension-keys --service-type RDS --identifier db-
LKCG0BK26374TPDFX0IWVCPM --start-time 1527026400 --end-time 1527080400 --
metric db.Load.avg --group-by '{"Group":"db.wait_event"}
```

Saída:

```
{
  "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
  }
]
}

```

- Para API obter detalhes, consulte [DescribeDimensionKeys](#) na Referência de AWS CLI Comandos.

get-resource-metrics

O código de exemplo a seguir mostra como usar `get-resource-metrics`.

AWS CLI

Para obter métricas de recursos

Este exemplo solicita pontos de dados para o grupo de dimensões `db.wait_event` e para a dimensão `db.wait_event.name` dentro desse grupo. Na resposta, os pontos de dados relevantes são agrupados pela dimensão solicitada (`db.wait_event.name`).

Comando:

```

aws pi get-resource-metrics --service-type RDS --identifier db-
LKCGOBK26374TPTDFX0IWVCPM --start-time 1527026400 --end-time 1527080400 --period-
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json

```

Os argumentos para `--metric-queries` são armazenados em um JSON arquivo, `metric-queries.json`. Veja o conteúdo desse arquivo:

```
[
  {
    "Metric": "db.load.avg",
    "GroupBy": {
      "Group": "db.wait_event"
    }
  }
]
```

Saída:

```
{
  "AlignedEndTime": 1.5270804E9,
  "AlignedStartTime": 1.5270264E9,
  "Identifier": "db-LKCG0BK26374TPTDFX0IWVCPM",
  "MetricList": [
    {
      "Key": {
        "Metric": "db.load.avg"
      },
      "DataPoints": [
        {
          "Timestamp": 1527026700.0,
          "Value": 1.3533333333333333
        },
        {
          "Timestamp": 1527027000.0,
          "Value": 0.88
        },
        <...remaining output omitted...>
      ]
    },
    {
      "Key": {
        "Metric": "db.load.avg",
        "Dimensions": {
          "db.wait_event.name": "wait/synch/mutex/innodb/aurora_lock_thread_slot_futex"
        }
      },
      "DataPoints": [
        {
          "Timestamp": 1527026700.0,
```

```
        "Value": 0.8566666666666667
      },
      {
        "Timestamp": 1527027000.0,
        "Value": 0.8633333333333333
      },
      <...remaining output omitted...>
    ],
  },
  <...remaining output omitted...>
]
}
```

- Para API obter detalhes, consulte [GetResourceMetrics](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Redshift usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Redshift.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-reserved-node-exchange

O código de exemplo a seguir mostra como usar `accept-reserved-node-exchange`.

AWS CLI

Para aceitar a troca de nós reservados

O `accept-reserved-node-exchange` exemplo a seguir aceita a troca de um nó DC1 reservado por um nó DC2 reservado.

```
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
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Atualização de nós reservados com o AWS CLI no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [AcceptReservedNodeExchange](#) na Referência de AWS CLI Comandos.

authorize-cluster-security-group-ingress

O código de exemplo a seguir mostra como usar `authorize-cluster-security-group-ingress`.

AWS CLI

Autorizar o acesso a um GroupThis exemplo EC2 de segurança autoriza o acesso a um grupo de EC2 segurança nomeado da Amazon.Comando:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

Autorizar o acesso a um CIDR rangeThis exemplo autoriza o acesso a um CIDR intervalo.Comando:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --cidrip 192.168.100.100/32
```

- Para API obter detalhes, consulte [AuthorizeClusterSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

authorize-snapshot-access

O código de exemplo a seguir mostra como usar `authorize-snapshot-access`.

AWS CLI

Autorizar uma AWS conta para restaurar um SnapshotThis exemplo autoriza a AWS conta a 444455556666 restaurar o snapshot. `my-snapshot-id` Por padrão, a saída está no JSON formato.Comando:

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

Resultado:

```
{
  "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"
  }
}

```

- Para API obter detalhes, consulte [AuthorizeSnapshotAccess](#) na Referência de AWS CLI Comandos.

batch-delete-cluster-snapshots

O código de exemplo a seguir mostra como usar `batch-delete-cluster-snapshots`.

AWS CLI

Para excluir um conjunto de instantâneos de cluster

O `batch-delete-cluster-snapshots` exemplo a seguir exclui um conjunto de instantâneos manuais do cluster.

```

aws redshift batch-delete-cluster-snapshots \
  --
  identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-

```

Saída:

```

{
  "Resources": [
    "mycluster-2019-11-06-14-12",

```

```
    "mycluster-2019-11-06-14-20"  
  ]  
}
```

Para obter mais informações, consulte [Amazon Redshift Snapshots](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [BatchDeleteClusterSnapshots](#) na Referência de AWS CLI Comandos.

batch-modify-cluster-snapshots

O código de exemplo a seguir mostra como usar `batch-modify-cluster-snapshots`.

AWS CLI

Para modificar um conjunto de instantâneos de cluster

O `batch-modify-cluster-snapshots` exemplo a seguir modifica as configurações de um conjunto de instantâneos de 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
```

Saída:

```
{  
  "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
}
}
```

Para obter mais informações, consulte [Amazon Redshift Snapshots](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [BatchModifyClusterSnapshots](#) na Referência de AWS CLI Comandos.

cancel-resize

O código de exemplo a seguir mostra como usar `cancel-resize`.

AWS CLI

Para cancelar o redimensionamento de um cluster

O `cancel-resize` exemplo a seguir cancela uma operação de redimensionamento clássica para um cluster.

```
aws redshift cancel-resize \  
  --cluster-identifier mycluster
```

Saída:

```
{  
  "TargetNodeType": "dc2.large",  
  "TargetNumberOfNodes": 2,  
  "TargetClusterType": "multi-node",  
  "Status": "CANCELLING",  
  "ResizeType": "ClassicResize",  
  "TargetEncryptionType": "NONE"  
}
```

Para obter mais informações, consulte [Redimensionamento de clusters no Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [CancelResize](#) na Referência de AWS CLI Comandos.

copy-cluster-snapshot

O código de exemplo a seguir mostra como usar `copy-cluster-snapshot`.

AWS CLI

O `copy-cluster-snapshot` obtém uma descrição de todo o cluster e retorna uma descrição de todas as versões do cluster. Por padrão, a saída está no JSON formatado. Comando:

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

Resultado:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CopyClusterSnapshot](#) na Referência de AWS CLI Comandos.

create-cluster-parameter-group

O código de exemplo a seguir mostra como usar `create-cluster-parameter-group`.

AWS CLI

Criar um GroupThis exemplo de parâmetro de cluster cria um novo grupo de parâmetros de cluster. Comando:

```
aws redshift create-cluster-parameter-group --parameter-group-name
myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
first cluster parameter group"
```

Resultado:

```
{
  "ClusterParameterGroup": {
    "ParameterGroupFamily": "redshift-1.0",
    "Description": "My first cluster parameter group",
    "ParameterGroupName": "myclusterparametergroup"
  },
  "ResponseMetadata": {
    "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
  }
}
```

- Para API obter detalhes, consulte [CreateClusterParameterGroup](#) na Referência de AWS CLI Comandos.

create-cluster-security-group

O código de exemplo a seguir mostra como usar create-cluster-security-group.

AWS CLI

A criação de um GroupThis exemplo de segurança de cluster cria um novo grupo de segurança de cluster. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group"
```

Resultado:

```
{
  "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"
    }
  }
}

```

Você pode obter as mesmas informações em formato de texto com a opção `--output text`.
Comando:

opção `--output text`. Comando:

opção. Comando:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group" --output text
```

Resultado:

```
This is my cluster security group 300454760768 mysecuritygroup
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- Para API obter detalhes, consulte [CreateClusterSecurityGroup](#) na Referência de AWS CLI Comandos.

create-cluster-snapshot

O código de exemplo a seguir mostra como usar `create-cluster-snapshot`.

AWS CLI

Criar um Snapshot Este exemplo de cluster cria um novo instantâneo do cluster. Por padrão, a saída está no JSON formato. Comando:


```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-identifier my-snapshot-id
```

Resultado:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateClusterSnapshot](#) na Referência de AWS CLI Comandos.

create-cluster-subnet-group

O código de exemplo a seguir mostra como usar `create-cluster-subnet-group`.

AWS CLI

Crie um GroupThis exemplo de sub-rede de cluster cria um novo grupo de sub-redes de cluster.Comando:

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup --description "My subnet group" --subnet-ids subnet-763fdd1c
```

Resultado:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateClusterSubnetGroup](#) na Referência de AWS CLI Comandos.

create-cluster

O código de exemplo a seguir mostra como usar `create-cluster`.

AWS CLI

O ParametersThis exemplo Criar um cluster com mínimo cria um cluster com o conjunto mínimo de parâmetros. Por padrão, a saída está no JSON formato.Comando:

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

Resultado:

```
{
  "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"
  }
}

```

- Para API obter detalhes, consulte [CreateCluster](#) na Referência de AWS CLI Comandos.

create-event-subscription

O código de exemplo a seguir mostra como usar `create-event-subscription`.

AWS CLI

Para criar uma assinatura de notificação para um evento

O `create-event-subscription` exemplo a seguir cria uma assinatura de notificação de eventos.

```
aws redshift create-event-subscription \  
  --subscription-name mysubscription \  
  --sns-topic-arn arn:aws:sns:us-west-2:123456789012:MySNSStopic \  
  --source-type cluster \  
  --source-ids mycluster
```

Saída:

```
{  
  "EventSubscription": {  
    "CustomerAwsId": "123456789012",  
    "CustSubscriptionId": "mysubscription",  
    "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNSStopic",  
    "Status": "active",  
    "SubscriptionCreationTime": "2019-12-09T20:05:19.365Z",  
    "SourceType": "cluster",  
    "SourceIdsList": [  
      "mycluster"  
    ],  
    "EventCategoriesList": [],  
    "Severity": "INFO",  
    "Enabled": true,  
    "Tags": []  
  }  
}
```

Para obter mais informações, consulte [Assinatura de notificações de eventos do Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [CreateEventSubscription](#) na Referência de AWS CLI Comandos.

create-hsm-client-certificate

O código de exemplo a seguir mostra como usar `create-hsm-client-certificate`.

AWS CLI

Para criar um certificado de HSM cliente

O `create-hsm-client-certificate` exemplo a seguir gera um certificado de HSM cliente que um cluster pode usar para se conectar a um HSM.

```
aws redshift create-hsm-client-certificate \
  --hsm-client-certificate-identifier myhsmclientcert
```

Saída:

```
{
  "HsmClientCertificate": {
    "HsmClientCertificateIdentifier": "myhsmclientcert",
    "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----
MIICiEXAMPLECQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMC
VVMxCzAJBgNVBAGTEXAMPLEEwDgYDVoQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC0lBTSBDb25EXAMPLEIwEAYDVoQDEwLUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD
EXAMPLETZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
b2x1MRIwEAEXAMPLEEwLUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
YXpvbi5jb20wgZ8wDQYJKEXAMPLEAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk6EXAMPLE3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugEXAMPLEezZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEEXAMPLEEAtCu4
nUhVVxYUEXAMPLEEh8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLE10ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
  }
}
```

Para obter mais informações, consulte a [Referência de API permissões do Amazon Redshift no Guia](#) de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [CreateHsmClientCertificate](#) na Referência de AWS CLI Comandos.

create-hsm-configuration

O código de exemplo a seguir mostra como usar create-hsm-configuration.

AWS CLI

Para criar uma HSM configuração

O `create-hsm-configuration` exemplo a seguir cria a HSM configuração especificada que contém as informações exigidas por um cluster para armazenar e usar chaves de criptografia de banco de dados em um módulo de segurança de 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
```

Saída:

```
{
  "HsmConfiguration": {
    "HsmConfigurationIdentifier": "myhsmconnection",
    "Description": "My HSM connection",
    "HsmIpAddress": "192.0.2.09",
    "HsmPartitionName": "myhsmpartition",
    "Tags": []
  }
}
```

- Para API obter detalhes, consulte [CreateHsmConfiguration](#) na Referência de AWS CLI Comandos.

create-snapshot-copy-grant

O código de exemplo a seguir mostra como usar `create-snapshot-copy-grant`.

AWS CLI

Para criar uma concessão de cópia de instantâneo

O `create-snapshot-copy-grant` exemplo a seguir cria uma concessão de cópia de instantâneo e criptografa instantâneos copiados em uma região de destino. AWS

```
aws redshift create-snapshot-copy-grant \
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

Saída:

```
{
  "SnapshotCopyGrant": {
    "SnapshotCopyGrantName": "mysnapshotcopygrantname",
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
    "Tags": []
  }
}
```

Para obter mais informações, consulte [Amazon Redshift Database Encryption](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [CreateSnapshotCopyGrant](#) na Referência de AWS CLI Comandos.

create-snapshot-schedule

O código de exemplo a seguir mostra como usar `create-snapshot-schedule`.

AWS CLI

Para criar uma agenda de instantâneos

O `create-snapshot-schedule` exemplo a seguir cria um cronograma de snapshot com a descrição especificada e uma taxa a cada 12 horas.

```
aws redshift create-snapshot-schedule \
  --schedule-definitions "rate(12 hours)" \
  --schedule-identifier mysnapshotschedule \
  --schedule-description "My schedule description"
```

Saída:

```
{
  "ScheduleDefinitions": [
    "rate(12 hours)"
  ],
  "ScheduleIdentifier": "mysnapshotschedule",
  "ScheduleDescription": "My schedule description",
  "Tags": []
}
```

```
}
```

Para obter mais informações, consulte [Programações automatizadas de snapshots no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [CreateSnapshotSchedule](#) na Referência de AWS CLI Comandos.

create-tags

O código de exemplo a seguir mostra como usar `create-tags`.

AWS CLI

Para criar tags para um cluster

O `create-tags` exemplo a seguir adiciona o par chave/valor da tag especificado ao cluster especificado.

```
aws redshift create-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tags "Key"="mytags", "Value"="tag1"
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Marcar recursos no Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [CreateTags](#) na Referência de AWS CLI Comandos.

delete-cluster-parameter-group

O código de exemplo a seguir mostra como usar `delete-cluster-parameter-group`.

AWS CLI

Excluir um GroupThis exemplo de parâmetro de cluster exclui um grupo de parâmetros de cluster. Comando:

```
aws redshift delete-cluster-parameter-group --parameter-group-name  
myclusterparametergroup
```


- Para API obter detalhes, consulte [DeleteClusterParameterGroup](#) na Referência de AWS CLI Comandos.

delete-cluster-security-group

O código de exemplo a seguir mostra como usar `delete-cluster-security-group`.

AWS CLI

Excluir um GroupThis exemplo de segurança de cluster exclui um grupo de segurança de cluster.Comando:

```
aws redshift delete-cluster-security-group --cluster-security-group-name
mysecuritygroup
```

- Para API obter detalhes, consulte [DeleteClusterSecurityGroup](#) na Referência de AWS CLI Comandos.

delete-cluster-snapshot

O código de exemplo a seguir mostra como usar `delete-cluster-snapshot`.

AWS CLI

Excluir um SnapshotThis exemplo de cluster exclui um snapshot do cluster.Comando:

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- Para API obter detalhes, consulte [DeleteClusterSnapshot](#) na Referência de AWS CLI Comandos.

delete-cluster-subnet-group

O código de exemplo a seguir mostra como usar `delete-cluster-subnet-group`.

AWS CLI

Excluir um GroupThis exemplo de sub-rede de cluster exclui um grupo de sub-redes de cluster.Comando:

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

Resultado:

```
{
  "ResponseMetadata": {
    "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
  }
}
```

- Para API obter detalhes, consulte [DeleteClusterSubnetGroup](#) na Referência de AWS CLI Comandos.

delete-cluster

O código de exemplo a seguir mostra como usar `delete-cluster`.

AWS CLI

O SnapshotThis exemplo Excluir um cluster sem cluster final exclui um cluster, forçando a exclusão de dados para que nenhum instantâneo final do cluster seja criado. Comando:

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

Excluir um cluster, Permitindo um SnapshotThis exemplo de cluster final exclui um cluster, mas especifica um instantâneo final do cluster. Comando:

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- Para API obter detalhes, consulte [DeleteCluster](#) na Referência de AWS CLI Comandos.

delete-event-subscription

O código de exemplo a seguir mostra como usar `delete-event-subscription`.

AWS CLI

Para excluir a assinatura do evento

O `delete-event-subscription` exemplo a seguir exclui a assinatura de notificação de eventos especificada.

```
aws redshift delete-event-subscription \  
  --subscription-name mysubscription
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Assinatura de notificações de eventos do Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DeleteEventSubscription](#) na Referência de AWS CLI Comandos.

delete-hsm-client-certificate

O código de exemplo a seguir mostra como usar `delete-hsm-client-certificate`.

AWS CLI

Para excluir o certificado HSM do cliente

O `delete-hsm-client-certificate` exemplo a seguir exclui um certificado de HSM cliente.

```
aws redshift delete-hsm-client-certificate \  
  --hsm-client-certificate-identifier myhsmclientcert
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte a [Referência de API permissões do Amazon Redshift no Guia](#) de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DeleteHsmClientCertificate](#) na Referência de AWS CLI Comandos.

delete-hsm-configuration

O código de exemplo a seguir mostra como usar `delete-hsm-configuration`.

AWS CLI

Para excluir uma HSM configuração

O `delete-hsm-configuration` exemplo a seguir exclui a HSM configuração especificada da AWS conta atual.

```
aws redshift delete-hsm-configuration /  
--hsm-configuration-identifier myhsmconnection
```

Esse comando não produz nenhuma saída.

- Para API obter detalhes, consulte [DeleteHsmConfiguration](#) na Referência de AWS CLI Comandos.

delete-scheduled-action

O código de exemplo a seguir mostra como usar `delete-scheduled-action`.

AWS CLI

Para excluir a ação agendada

O `delete-scheduled-action` exemplo a seguir exclui a ação agendada especificada.

```
aws redshift delete-scheduled-action \  
--scheduled-action-name myscheduledaction
```

Esse comando não produz nenhuma saída.

- Para API obter detalhes, consulte [DeleteScheduledAction](#) na Referência de AWS CLI Comandos.

delete-snapshot-copy-grant

O código de exemplo a seguir mostra como usar `delete-snapshot-copy-grant`.

AWS CLI

Para excluir a concessão de cópia de instantâneo

O `delete-snapshot-copy-grant` exemplo a seguir exclui a concessão de cópia de instantâneo especificada.

```
aws redshift delete-snapshot-copy-grant \  

```

```
--snapshot-copy-grant-name mysnapshotcopygrantname
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Amazon Redshift Database Encryption](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [DeleteSnapshotCopyGrant](#) na Referência de AWS CLI Comandos.

delete-snapshot-schedule

O código de exemplo a seguir mostra como usar delete-snapshot-schedule.

AWS CLI

Para excluir a programação de snapshots

O delete-snapshot-schedule exemplo a seguir exclui a programação de snapshot especificada. Você deve desassociar os clusters antes de excluir a agenda.

```
aws redshift delete-snapshot-schedule \  
--schedule-identifier mysnapshotschedule
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Programações automatizadas de snapshots no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [DeleteSnapshotSchedule](#) na Referência de AWS CLI Comandos.

delete-tags

O código de exemplo a seguir mostra como usar delete-tags.

AWS CLI

Para excluir tags de um cluster

O delete-tags exemplo a seguir exclui as tags com os nomes de chave especificados do cluster especificado.

```
aws redshift delete-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tag-keys "clustertagkey" "clustertagvalue"
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Marcar recursos no Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DeleteTags](#) na Referência de AWS CLI Comandos.

describe-account-attributes

O código de exemplo a seguir mostra como usar describe-account-attributes.

AWS CLI

Para descrever os atributos de uma AWS conta

O describe-account-attributes exemplo a seguir exibe os atributos associados à AWS conta de chamada.

```
aws redshift describe-account-attributes
```

Saída:

```
{  
  "AccountAttributes": [  
    {  
      "AttributeName": "max-defer-maintenance-duration",  
      "AttributeValues": [  
        {  
          "AttributeValue": "45"  
        }  
      ]  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeAccountAttributes](#) na Referência de AWS CLI Comandos.

describe-cluster-db-revisions

O código de exemplo a seguir mostra como usar `describe-cluster-db-revisions`.

AWS CLI

Para descrever as revisões de banco de dados para um cluster

O `describe-cluster-db-revisions` exemplo a seguir exibe os detalhes de uma matriz de `ClusterDbRevision` objetos para o cluster especificado.

```
aws redshift describe-cluster-db-revisions \  
  --cluster-identifier mycluster
```

Saída:

```
{  
  "ClusterDbRevisions": [  
    {  
      "ClusterIdentifier": "mycluster",  
      "CurrentDatabaseRevision": "11420",  
      "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",  
      "RevisionTargets": []  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DescribeClusterDbRevisions](#) na Referência de AWS CLI Comandos.

describe-cluster-parameter-groups

O código de exemplo a seguir mostra como usar `describe-cluster-parameter-groups`.

AWS CLI

O GroupsThis exemplo Obter uma descrição de todos os parâmetros do cluster retorna uma descrição de todos os grupos de parâmetros do cluster da conta, com cabeçalhos de coluna. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-cluster-parameter-groups
```

Resultado:

```
{
  "ParameterGroups": [
    {
      "ParameterGroupFamily": "redshift-1.0",
      "Description": "My first cluster parameter group",
      "ParameterGroupName": "myclusterparametergroup"
    } ],
  "ResponseMetadata": {
    "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
  }
}
```

Você pode obter as mesmas informações em formato de texto com a opção `--output text`.
Comando:

opção `--output text`. Comando:

opção. Comando:

```
aws redshift describe-cluster-parameter-groups --output text
```

Resultado:

```
redshift-1.0      My first cluster parameter group      myclusterparametergroup
RESPONSEMETADATA 9e665a36-64cc-11e2-8f7d-3b939af52818
```

- Para API obter detalhes, consulte [DescribeClusterParameterGroups](#) na Referência de AWS CLI Comandos.

describe-cluster-parameters

O código de exemplo a seguir mostra como usar `describe-cluster-parameters`.

AWS CLI

O GroupThis exemplo de Recuperar os parâmetros para um parâmetro de cluster especificado recupera os parâmetros do grupo de parâmetros nomeado. Por padrão, a saída está no JSON formato. Comando:


```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup
```

Resultado:

```
{
  "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...)
  ]
}
```

Você pode obter as mesmas informações em formato de texto com a opção `--output text`.

Comando:

opção `--output text`. Comando:

opção. Comando:

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup --output text
```

Resultado:

```
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

```

- Para API obter detalhes, consulte [DescribeClusterParameters](#) na Referência de AWS CLI Comandos.

describe-cluster-security-groups

O código de exemplo a seguir mostra como usar `describe-cluster-security-groups`.

AWS CLI

O GroupsThis exemplo Obter uma descrição de toda a segurança do cluster retorna uma descrição de todos os grupos de segurança do cluster da conta. Por padrão, a saída está no JSON formato.Comando:

```
aws redshift describe-cluster-security-groups
```

Resultado:

```
{
  "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...)
]
}

```

- Para API obter detalhes, consulte [DescribeClusterSecurityGroups](#) na Referência de AWS CLI Comandos.

describe-cluster-snapshots

O código de exemplo a seguir mostra como usar `describe-cluster-snapshots`.

AWS CLI

O SnapshotsThis exemplo Obter uma descrição de todo o cluster retorna uma descrição de todos os instantâneos do cluster da conta. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-cluster-snapshots
```

Resultado:

```

{
  "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...)
```

- Para API obter detalhes, consulte [DescribeClusterSnapshots](#) na Referência de AWS CLI Comandos.

describe-cluster-subnet-groups

O código de exemplo a seguir mostra como usar `describe-cluster-subnet-groups`.

AWS CLI

O GroupsThis exemplo Obter uma descrição de todas as sub-redes do cluster retorna uma descrição de todos os grupos de sub-redes do cluster. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-cluster-subnet-groups
```

Resultado:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [DescribeClusterSubnetGroups](#) na Referência de AWS CLI Comandos.

describe-cluster-tracks

O código de exemplo a seguir mostra como usar `describe-cluster-tracks`.

AWS CLI

Para descrever as trilhas do cluster

O `describe-cluster-tracks` exemplo a seguir exibe detalhes das faixas de manutenção disponíveis.

```
aws redshift describe-cluster-tracks \  
  --maintenance-track-name current
```

Saída:

```
{  
  "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"  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

```

    ]
  }
]
}

```

Para obter mais informações, consulte [Escolhendo faixas de manutenção de clusters](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeClusterTracks](#) na Referência de AWS CLI Comandos.

describe-cluster-versions

O código de exemplo a seguir mostra como usar `describe-cluster-versions`.

AWS CLI

O VersionsThis exemplo Obter uma descrição de todo o cluster retorna uma descrição de todas as versões do cluster. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-cluster-versions
```

Resultado:

```

{
  "ClusterVersions": [
    {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
    } ],
  "ResponseMetadata": {
    "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
  }
}

```

- Para API obter detalhes, consulte [DescribeClusterVersions](#) na Referência de AWS CLI Comandos.

describe-clusters

O código de exemplo a seguir mostra como usar `describe-clusters`.

AWS CLI

O `ClustersThis` exemplo Obter uma descrição de tudo retorna uma descrição de todos os clusters da conta. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-clusters
```

Resultado:

```
{
  "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"
  }
}

```

Você pode obter as mesmas informações em formato de texto com a opção `--output text`.
Comando:

opção `--output text`. Comando:

opção. Comando:

```
aws redshift describe-clusters --output text
```

Resultado:

```

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

```

- Para API obter detalhes, consulte [DescribeClusters](#) na Referência de AWS CLI Comandos.

describe-default-cluster-parameters

O código de exemplo a seguir mostra como usar `describe-default-cluster-parameters`.

AWS CLI

O ParametersThis exemplo Obter uma descrição do cluster padrão retorna uma descrição dos parâmetros de cluster padrão para a `redshift-1.0` família. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-default-cluster-parameters --parameter-group-family
redshift-1.0
```

Resultado:

```
{
  "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...)
    ]
  }
}
```

Para ver uma lista de famílias de grupos de parâmetros válidas, use o `describe-cluster-parameter-groups` comando.

`describe-cluster-parameter-groups` comando.

comando.

- Para API obter detalhes, consulte [DescribeDefaultClusterParameters](#) na Referência de AWS CLI Comandos.

describe-event-categories

O código de exemplo a seguir mostra como usar `describe-event-categories`.

AWS CLI

Para descrever categorias de eventos para um cluster

O `describe-event-categories` exemplo a seguir exibe detalhes das categorias de eventos de um cluster.

```
aws redshift describe-event-categories \  
  --source-type cluster
```

Saída:

```
{  
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [DescribeEventCategories](#) na Referência de AWS CLI Comandos.

describe-event-subscriptions

O código de exemplo a seguir mostra como usar `describe-event-subscriptions`.

AWS CLI

Para descrever as assinaturas de eventos

O `describe-event-subscriptions` exemplo a seguir exibe assinaturas de notificação de eventos para a assinatura especificada.

```
aws redshift describe-event-subscriptions \
  --subscription-name mysubscription
```

Saída:

```

{
  "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": []
    }
]
}

```

Para obter mais informações, consulte [Assinatura de notificações de eventos do Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeEventSubscriptions](#) na Referência de AWS CLI Comandos.

describe-events

O código de exemplo a seguir mostra como usar `describe-events`.

AWS CLI

Descreva todos os eventos Este exemplo retorna todos os eventos. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-events
```

Resultado:

```

{
  "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"
  }
}

```

Você pode obter as mesmas informações em formato de texto com a opção `--output text`.
Comando:

opção `--output text`. Comando:

opção. Comando:

```
aws redshift describe-events --output text
```

Resultado:

```
2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
myclusterparametergroup has been created.    cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17
```

- Para API obter detalhes, consulte [DescribeEvents](#) na Referência de AWS CLI Comandos.

describe-hsm-client-certificates

O código de exemplo a seguir mostra como usar `describe-hsm-client-certificates`.

AWS CLI

Para descrever os certificados HSM do cliente

O `describe-hsm-client-certificates` exemplo a seguir exibe detalhes do certificado de HSM cliente especificado.

```
aws redshift describe-hsm-client-certificates \
--hsm-client-certificate-identifier myhsmclientcert
```

Saída:

```
{
  "HsmClientCertificates": [
    {
      "HsmClientCertificateIdentifier": "myhsmclientcert",
      "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\
EXAMPLECAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC01BTSBDb25zEXAMPLEwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGfYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN
EXAMPLE0MjA0NTIxwJCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAGTA1dBMRAdDgYD
VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
```

```

    b2x1MRIwEAYDVQDEw1UZxN0Q21sEXAMPLEdBgkqhkiG9w0BCQEWEG5vb251QGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEXAMPLEMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY67EXAMPLEE
    EXAMPLEZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9EXAMPLE6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJI1J00zbhNYS5f6GuoEDEXAMPLEBHjJnyp3780D8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
  }
]
}

```

Para obter mais informações, consulte a [Referência de API permissões do Amazon Redshift no Guia](#) de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeHsmClientCertificates](#) na Referência de AWS CLI Comandos.

describe-hsm-configurations

O código de exemplo a seguir mostra como usar `describe-hsm-configurations`.

AWS CLI

Para descrever as HSM configurações

O `describe-hsm-configurations` exemplo a seguir exibe detalhes das HSM configurações disponíveis para a AWS conta de chamada.

```

aws redshift describe-hsm-configurations /
--hsm-configuration-identifier myhsmconnection

```

Saída:

```

{
  "HsmConfigurations": [
    {
      "HsmConfigurationIdentifier": "myhsmconnection",
      "Description": "My HSM connection",
      "HsmIpAddress": "192.0.2.09",
      "HsmPartitionName": "myhsmpartition",
    }
  ]
}

```

```
    "Tags": []
  }
]
}
```

- Para API obter detalhes, consulte [DescribeHsmConfigurations](#) na Referência de AWS CLI Comandos.

describe-logging-status

O código de exemplo a seguir mostra como usar `describe-logging-status`.

AWS CLI

Para descrever o status de registro de um cluster

O `describe-logging-status` exemplo a seguir mostra se as informações, como consultas e tentativas de conexão, estão sendo registradas em um cluster.

```
aws redshift describe-logging-status \
  --cluster-identifier mycluster
```

Saída:

```
{
  "LoggingEnabled": false
}
```

Para obter mais informações, consulte [Registro de auditoria de banco](#) de dados no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeLoggingStatus](#) na Referência de AWS CLI Comandos.

describe-node-configuration-options

O código de exemplo a seguir mostra como usar `describe-node-configuration-options`.

AWS CLI

Para descrever as opções de configuração do nó

O `describe-node-configuration-options` exemplo a seguir exibe as propriedades de possíveis configurações de nós, como tipo de nó, número de nós e uso do disco para o snapshot de cluster especificado.

```
aws redshift describe-node-configuration-options \  
  --action-type restore-cluster \  
  --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Comprar nós reservados do Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeNodeConfigurationOptions](#) na Referência de AWS CLI Comandos.

describe-orderable-cluster-options

O código de exemplo a seguir mostra como usar `describe-orderable-cluster-options`.

AWS CLI

O `OptionsThis` exemplo de descrição de todos os clusters solicitáveis retorna descrições de todas as opções de clusters solicitáveis. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift describe-orderable-cluster-options
```

Resultado:

```
{
  "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"
  }
}

```

Você pode obter as mesmas informações em formato de texto com a opção `--output text`.
Comando:

opção `--output text`. Comando:

opção. Comando:

```
aws redshift describe-orderable-cluster-options --output text
```

Resultado:

```

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

```

- Para API obter detalhes, consulte [DescribeOrderableClusterOptions](#) na Referência de AWS CLI Comandos.

describe-reserved-node-offerings

O código de exemplo a seguir mostra como usar `describe-reserved-node-offerings`.

AWS CLI

O OfferingsThis exemplo de descrição do nó reservado mostra todas as ofertas de nós reservados que estão disponíveis para compra. Comando:

```
aws redshift describe-reserved-node-offerings
```

Resultado:

```
{
  "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 você quiser comprar uma oferta de nó reservado, você pode ligar `purchase-reserved-node-offering` usando um válido `ReservedNodeOfferingId`.

`purchase-reserved-node-offering` usando um válido `ReservedNodeOfferingId`.

usando um válido `ReservedNodeOfferingId`.

`ReservedNodeOfferingId`.

.

- Para API obter detalhes, consulte [DescribeReservedNodeOfferings](#) na Referência de AWS CLI Comandos.

describe-reserved-nodes

O código de exemplo a seguir mostra como usar `describe-reserved-nodes`.

AWS CLI

Descreva Reservado O Nodes This exemplo mostra uma oferta de nó reservado que foi comprada. Comando:

```
aws redshift describe-reserved-nodes
```

Resultado:

```
{
  "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"
    }
]
}
```

- Para API obter detalhes, consulte [DescribeReservedNodes](#) na Referência de AWS CLI Comandos.

describe-resize

O código de exemplo a seguir mostra como usar `describe-resize`.

AWS CLI

O `ResizeThis` exemplo descreve o último redimensionamento de um cluster. A solicitação foi para 3 nós do tipo `dw.hs1.8xlarge`. `Command`:

```
aws redshift describe-resize --cluster-identifier mycluster
```

Resultado:

```
{
  "Status": "NONE",
  "TargetClusterType": "multi-node",
  "TargetNodeType": "dw.hs1.8xlarge",
  "ResponseMetadata": {
    "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"
  },
  "TargetNumberOfNodes": "3"
}
```

- Para API obter detalhes, consulte [DescribeResize](#) na Referência de AWS CLI Comandos.

describe-scheduled-actions

O código de exemplo a seguir mostra como usar `describe-scheduled-actions`.

AWS CLI

Para descrever ações agendadas

O `describe-scheduled-actions` exemplo a seguir exibe detalhes de todas as ações programadas atualmente.

```
aws redshift describe-scheduled-actions
```

Saída:

```
{
  "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"
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeScheduledActions](#) na Referência de AWS CLI Comandos.

describe-snapshot-copy-grants

O código de exemplo a seguir mostra como usar `describe-snapshot-copy-grants`.

AWS CLI

Para descrever as concessões de cópia instantânea

O `describe-snapshot-copy-grants` exemplo a seguir exibe detalhes da concessão de cópia de snapshot de cluster especificada.

```
aws redshift describe-snapshot-copy-grants \  
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

Saída:

```
{  
  "SnapshotCopyGrants": [  
    {  
      "SnapshotCopyGrantName": "mysnapshotcopygrantname",  
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/  
bPxRfih3yCo8nvbEXAMPLEKEY",  
      "Tags": []  
    }  
  ]  
}
```

Para obter mais informações, consulte [Amazon Redshift Database Encryption](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [DescribeSnapshotCopyGrants](#) na Referência de AWS CLI Comandos.

describe-snapshot-schedules

O código de exemplo a seguir mostra como usar `describe-snapshot-schedules`.

AWS CLI

Para descrever as programações de snapshots

O `describe-snapshot-schedules` exemplo a seguir exibe detalhes da programação de snapshot de cluster especificada.

```
aws redshift describe-snapshot-schedules \  
  --cluster-identifier mycluster \  
  --schedule-identifier mysnapshotschedule
```

Saída:

```
{
```



```
"SnapshotSchedules": [
  {
    "ScheduleDefinitions": [
      "rate(12 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": [],
    "AssociatedClusterCount": 1,
    "AssociatedClusters": [
      {
        "ClusterIdentifier": "mycluster",
        "ScheduleAssociationState": "ACTIVE"
      }
    ]
  }
]
```

Para obter mais informações, consulte [Programações automatizadas de snapshots no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [DescribeSnapshotSchedules](#) na Referência de AWS CLI Comandos.

describe-storage

O código de exemplo a seguir mostra como usar `describe-storage`.

AWS CLI

Para descrever o armazenamento

O `describe-storage` exemplo a seguir exibe detalhes sobre o armazenamento de backup e os tamanhos de armazenamento provisório da conta.

```
aws redshift describe-storage
```

Saída:

```
{
```

```
"TotalBackupSizeInMegaBytes": 193149.0,  
"TotalProvisionedStorageInMegaBytes": 655360.0  
}
```

Para obter mais informações, consulte [Gerenciando o armazenamento de snapshots no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [DescribeStorage](#) na Referência de AWS CLI Comandos.

describe-table-restore-status

O código de exemplo a seguir mostra como usar `describe-table-restore-status`.

AWS CLI

Para descrever o status das solicitações de restauração de tabela a partir de um snapshot de cluster

O `describe-table-restore-status` exemplo a seguir exibe detalhes das solicitações de restauração de tabela feitas para o cluster especificado.

```
aws redshift describe-table-restore-status /  
--cluster-identifier mycluster
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
}
```

Para obter mais informações, consulte [Restauração de uma tabela a partir de um snapshot no Guia](#) de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeTableRestoreStatus](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever as tags

O `describe-tags` exemplo a seguir exibe os recursos que o cluster especificado associou aos nomes e valores das tags especificados.

```
aws redshift describe-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tag-keys clustertagkey \  
  --tag-values clustertagvalue
```

Saída:

```
{  
  "TaggedResources": [  
    {  
      "Tag": {  
        "Key": "clustertagkey",  
        "Value": "clustertagvalue"  
      },  
      "ResourceName": "arn:aws:redshift:us-  
west-2:123456789012:cluster:mycluster",  
      "ResourceType": "cluster"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Marcar recursos no Amazon](#) Redshift no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

disable-snapshot-copy

O código de exemplo a seguir mostra como usar `disable-snapshot-copy`.

AWS CLI

Para desativar a cópia instantânea de um cluster

O `disable-snapshot-copy` exemplo a seguir desativa a cópia automática de um snapshot para o cluster especificado.

```
aws redshift disable-snapshot-copy \  
  --cluster-identifier mycluster
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Cópia de snapshots para outra AWS região](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [DisableSnapshotCopy](#) na Referência de AWS CLI Comandos.

enable-snapshot-copy

O código de exemplo a seguir mostra como usar `enable-snapshot-copy`.

AWS CLI

Para habilitar a cópia instantânea para um cluster

O `enable-snapshot-copy` exemplo a seguir permite a cópia automática de um snapshot para o cluster especificado.

```
aws redshift enable-snapshot-copy \  
  --cluster-identifier mycluster \  
  --destination-region us-west-1
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Cópia de snapshots para outra AWS região](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [EnableSnapshotCopy](#) na Referência de AWS CLI Comandos.

get-cluster-credentials

O código de exemplo a seguir mostra como usar `get-cluster-credentials`.

AWS CLI

Para obter credenciais de cluster para uma conta AWS

O `get-cluster-credentials` exemplo a seguir recupera credenciais temporárias que permitem o acesso a um banco de dados do Amazon Redshift.

```
aws redshift get-cluster-credentials \  
  --db-user adminuser --db-name dev \  
  --cluster-identifier mycluster
```

Saída:

```
{  
  "DbUser": "IAM:adminuser",  
  "DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx109d6UmpkBtvEqFly/  
EXAMPLE==",  
  "Expiration": "2019-12-10T17:25:05.770Z"  
}
```

Para obter mais informações, consulte [Geração IAM de credenciais de banco de dados usando o Amazon CLI Redshift API](#) ou no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [GetClusterCredentials](#) na Referência de AWS CLI Comandos.

get-reserved-node-exchange-offerings

O código de exemplo a seguir mostra como usar `get-reserved-node-exchange-offerings`.

AWS CLI

Para obter ofertas reservadas de troca de nós

O `get-reserved-node-exchange-offerings` exemplo a seguir recupera uma matriz DC2 `ReservedNodeOfferings` que corresponde ao nó DC1 reservado especificado.

```
aws redshift get-reserved-node-exchange-offerings \
  --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Atualização de nós reservados com o AWS CLI no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [GetReservedNodeExchangeOfferings](#) na Referência de AWS CLI Comandos.

modify-cluster-iam-roles

O código de exemplo a seguir mostra como usar `modify-cluster-iam-roles`.

AWS CLI

Para modificar a IAM função de um cluster

O `modify-cluster-iam-roles` exemplo a seguir remove a AWS IAM função especificada do cluster especificado.

```
aws redshift modify-cluster-iam-roles \  
  --cluster-identifier mycluster \  
  --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Usando políticas \(IAMpolíticas\) baseadas em identidade para o Amazon Redshift no Guia](#) de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [ModifyClusterIamRoles](#) na Referência de AWS CLI Comandos.

modify-cluster-maintenance

O código de exemplo a seguir mostra como usar `modify-cluster-maintenance`.

AWS CLI

Para modificar a manutenção do cluster

O `modify-cluster-maintenance` exemplo a seguir adia a manutenção do cluster especificado em 30 dias.

```
aws redshift modify-cluster-maintenance \  
  --cluster-identifier mycluster \  
  --defer-maintenance \  
  --defer-maintenance-duration 30
```

Saída:

```
{  
  "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-mUdVIffFcT1B4SGhw6fyF",
            "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
            "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
        }
    ],
    "ExpectedNextSnapshotScheduleTime": "2019-12-11T04:42:55.631Z",
    "ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
    "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z"
}
}

```

Para obter mais informações, consulte [Manutenção de clusters](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [ModifyClusterMaintenance](#) na Referência de AWS CLI Comandos.

modify-cluster-parameter-group

O código de exemplo a seguir mostra como usar `modify-cluster-parameter-group`.

AWS CLI

Modificar um parâmetro em um grupo de parâmetros

O `modify-cluster-parameter-group` exemplo a seguir modifica o parâmetro `wlm_json_configuration` para o gerenciamento da carga de trabalho. Ele aceita os parâmetros de um arquivo que contém o JSON conteúdo mostrado abaixo.

```
aws redshift modify-cluster-parameter-group \  
  --parameter-group-name myclusterparametergroup \  
  --parameters file://modify_pg.json
```

Conteúdo de `modify_pg.json`:

```
[  
  {  
    "ParameterName": "wlm_json_configuration",  
    "ParameterValue": "[{\"user_group\":\"example_user_group1\",\"query_group\":  
  \"example_query_group1\", \"query_concurrency\":7},{\"query_concurrency\":5}]"  
  }  
]
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [ModifyClusterParameterGroup](#) na Referência de AWS CLI Comandos.

modify-cluster-snapshot-schedule

O código de exemplo a seguir mostra como usar `modify-cluster-snapshot-schedule`.

AWS CLI

Para modificar o cronograma de snapshots do cluster

O `modify-cluster-snapshot-schedule` exemplo a seguir remove a programação de snapshot especificada do cluster especificado.

```
aws redshift modify-cluster-snapshot-schedule \  
  --cluster-identifier mycluster \  
  --schedule-identifier mynapshotschedule \  
  --disassociate-schedule
```

Esse comando não produz nenhuma saída.

Para obter mais informações, consulte [Programações automatizadas de snapshots no Guia de gerenciamento de clusters do Amazon Redshift](#).

- Para API obter detalhes, consulte [ModifyClusterSnapshotSchedule](#) na Referência de AWS CLI Comandos.

modify-cluster-snapshot

O código de exemplo a seguir mostra como usar `modify-cluster-snapshot`.

AWS CLI

Para modificar o instantâneo do cluster

O `modify-cluster-snapshot` exemplo a seguir define a configuração do período de retenção manual para o snapshot de cluster especificado no valor de 10 dias.

```
aws redshift modify-cluster-snapshot \  
  --snapshot-identifier mycluster-2019-11-06-16-32 \  
  --manual-snapshot-retention-period 10
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Amazon Redshift Snapshots](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [ModifyClusterSnapshot](#) na Referência de AWS CLI Comandos.

modify-cluster-subnet-group

O código de exemplo a seguir mostra como usar `modify-cluster-subnet-group`.

AWS CLI

O GroupThis exemplo de Modificar as sub-redes em uma sub-rede de cluster mostra como modificar a lista de sub-redes em um grupo de sub-redes de cache. Por padrão, a saída está no JSON formato.Comando:

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
--subnet-ids subnet-763fdd1 subnet-ac830e9
```

Resultado:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [ModifyClusterSubnetGroup](#) na Referência de AWS CLI Comandos.

modify-cluster

O código de exemplo a seguir mostra como usar `modify-cluster`.

AWS CLI

Associar um grupo de segurança a um ClusterThis exemplo mostra como associar um grupo de segurança de cluster ao cluster especificado. Comando:

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups mysecuritygroup
```

Modificar a janela de manutenção para a ClusterThis mostra como alterar a janela de manutenção semanal preferencial de um cluster para a janela mínima de quatro horas, começando aos domingos às 23h15 e terminando às segundas-feiras às 3h15. Comando:

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-window Sun:23:15-Mon:03:15
```

Alterar a senha mestra do ClusterThis exemplo mostra como alterar a senha mestra de um cluster. Comando:

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password A1b2c3d4
```

- Para API obter detalhes, consulte [ModifyCluster](#) na Referência de AWS CLI Comandos.

modify-event-subscription

O código de exemplo a seguir mostra como usar `modify-event-subscription`.

AWS CLI

Para modificar a assinatura do evento

O `modify-event-subscription` exemplo a seguir desativa a assinatura de notificação de eventos especificada.

```
aws redshift modify-event-subscription \
```

```
--subscription-name mysubscription \  
--no-enabled
```

Saída:

```
{  
  "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": []  
  }  
}
```

Para obter mais informações, consulte [Assinatura de notificações de eventos do Amazon Redshift](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [ModifyEventSubscription](#) na Referência de AWS CLI Comandos.

modify-scheduled-action

O código de exemplo a seguir mostra como usar `modify-scheduled-action`.

AWS CLI

Para modificar a ação agendada

O `modify-scheduled-action` exemplo a seguir adiciona uma descrição à ação agendada existente especificada.

```
aws redshift modify-scheduled-action \  
--scheduled-action-name myscheduledaction \  
--description mydescription
```

```
--scheduled-action-description "My scheduled action"
```

Saída:

```
{
  "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"
  ]
}
```

- Para API obter detalhes, consulte [ModifyScheduledAction](#) na Referência de AWS CLI Comandos.

modify-snapshot-copy-retention-period

O código de exemplo a seguir mostra como usar `modify-snapshot-copy-retention-period`.

AWS CLI

Para modificar o período de retenção da cópia instantânea

O `modify-snapshot-copy-retention-period` exemplo a seguir modifica o número de dias para reter os instantâneos do cluster especificado na AWS região de destino após serem copiados da região de origem. AWS

```
aws redshift modify-snapshot-copy-retention-period \  
  --cluster-identifier mycluster \  
  --retention-period 15
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Snapshot Schedule Format](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [ModifySnapshotCopyRetentionPeriod](#) na Referência de AWS CLI Comandos.

modify-snapshot-schedule

O código de exemplo a seguir mostra como usar `modify-snapshot-schedule`.

AWS CLI

Para modificar a programação de snapshots

O `modify-snapshot-schedule` exemplo a seguir modifica a taxa da programação de snapshot especificada para cada 10 horas.

```
aws redshift modify-snapshot-schedule \
```

```
--schedule-identifier mynapshotschedule \  
--schedule-definitions "rate(10 hours)"
```

Saída:

```
{  
  "ScheduleDefinitions": [  
    "rate(10 hours)"  
  ],  
  "ScheduleIdentifier": "mynapshotschedule",  
  "ScheduleDescription": "My schedule description",  
  "Tags": []  
}
```

Para obter mais informações, consulte [Snapshot Schedule Format](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [ModifySnapshotSchedule](#) na Referência de AWS CLI Comandos.

purchase-reserved-node-offering

O código de exemplo a seguir mostra como usar `purchase-reserved-node-offering`.

AWS CLI

O NodeThis exemplo de compra de um nó reservado mostra como comprar uma oferta de nó reservado. O `reserved-node-offering-id` é obtido chamando `describe-reserved-node-offerings`. Command:

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-  
cf4c-4343-be8b-d832c45ab51c
```

Resultado:

```
{  
  "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"
  }
}

```

- Para API obter detalhes, consulte [PurchaseReservedNodeOffering](#) na Referência de AWS CLI Comandos.

reboot-cluster

O código de exemplo a seguir mostra como usar `reboot-cluster`.

AWS CLI

Reinicialize um Cluster This exemplo reinicializa um cluster. Por padrão, a saída está no JSON formato. Comando:

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

Resultado:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [RebootCluster](#) na Referência de AWS CLI Comandos.

reset-cluster-parameter-group

O código de exemplo a seguir mostra como usar `reset-cluster-parameter-group`.

AWS CLI

Redefinir parâmetros em um GroupThis exemplo de parâmetro mostra como redefinir todos os parâmetros em um grupo de parâmetros. Comando:

```
aws redshift reset-cluster-parameter-group --parameter-group-name
myclusterparametergroup --reset-all-parameters
```

- Para API obter detalhes, consulte [ResetClusterParameterGroup](#) na Referência de AWS CLI Comandos.

resize-cluster

O código de exemplo a seguir mostra como usar `resize-cluster`.

AWS CLI

Para redimensionar o cluster

O `resize-cluster` exemplo a seguir redimensiona o cluster especificado.

```
aws redshift resize-cluster \  
  --cluster-identifier mycluster \  
  --cluster-type multi-node \  
  --node-type dc2.large \  
  --number-of-nodes 6 \  
  --classic
```

Saída:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "resizing",  
    "ClusterAvailabilityStatus": "Modifying",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswqeuae.us-west-2.redshift.amazonaws.com",  
      "Port": 5439  
    },  
    "ClusterCreateTime": "2019-12-05T18:44:36.991Z",  
    "AutomatedSnapshotRetentionPeriod": 3,  
    "ManualSnapshotRetentionPeriod": -1,  
    "ClusterSecurityGroups": [],  
    "VpcSecurityGroups": [  

```

```
{
  "VpcSecurityGroupId": "sh-a1a123ab",
  "Status": "active"
},
"ClusterParameterGroups": [
  {
    "ParameterGroupName": "default.redshift-1.0",
    "ParameterApplyStatus": "in-sync"
  }
],
"ClusterSubnetGroupName": "default",
"VpcId": "vpc-a1abc1a1",
"AvailabilityZone": "us-west-2f",
"PreferredMaintenanceWindow": "sat:16:00-sat:16:30",
"PendingModifiedValues": {
  "NodeType": "dc2.large",
  "NumberOfNodes": 6,
  "ClusterType": "multi-node"
},
"ClusterVersion": "1.0",
"AllowVersionUpgrade": true,
"NumberOfNodes": 4,
"PubliclyAccessible": false,
"Encrypted": false,
"ClusterSnapshotCopyStatus": {
  "DestinationRegion": "us-west-1",
  "RetentionPeriod": 15,
  "ManualSnapshotRetentionPeriod": -1
},
"Tags": [
  {
    "Key": "mytags",
    "Value": "tag1"
  }
],
"EnhancedVpcRouting": false,
"IamRoles": [],
"MaintenanceTrackName": "current",
"DeferredMaintenanceWindows": [
  {
    "DeferMaintenanceIdentifier": "dfm-mUdVCfDcT1B4SGhw6fyF",
    "DeferMaintenanceStartTime": "2019-12-10T18:18:39.354Z",
    "DeferMaintenanceEndTime": "2020-01-09T18:18:39.354Z"
  }
]
```

```

    }
  ],
  "NextMaintenanceWindowStartTime": "2020-01-11T16:00:00Z",
  "ResizeInfo": {
    "ResizeType": "ClassicResize",
    "AllowCancelResize": true
  }
}
}
}

```

Para obter mais informações, consulte Como [redimensionar um cluster](#) no Guia de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [ResizeCluster](#) na Referência de AWS CLI Comandos.

restore-from-cluster-snapshot

O código de exemplo a seguir mostra como usar `restore-from-cluster-snapshot`.

AWS CLI

Restaurar um cluster A partir de um Snapshot This exemplo restaura um cluster a partir de um snapshot. Comando:

```
aws redshift restore-from-cluster-snapshot --cluster-identifier mycluster-clone --snapshot-identifier my-snapshot-id
```

Resultado:

```

{
  "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"
}
}

```

- Para API obter detalhes, consulte [RestoreFromClusterSnapshot](#) na Referência de AWS CLI Comandos.

restore-table-from-cluster-snapshot

O código de exemplo a seguir mostra como usar `restore-table-from-cluster-snapshot`.

AWS CLI

Para restaurar a tabela a partir de um snapshot de cluster

O `restore-table-from-cluster-snapshot` exemplo a seguir cria uma nova tabela a partir da tabela especificada no snapshot do cluster especificado.

```

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
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Restauração de uma tabela a partir de um snapshot no Guia](#) de gerenciamento de clusters do Amazon Redshift.

- Para API obter detalhes, consulte [RestoreTableFromClusterSnapshot](#) na Referência de AWS CLI Comandos.

revoke-cluster-security-group-ingress

O código de exemplo a seguir mostra como usar `revoke-cluster-security-group-ingress`.

AWS CLI

Revogar o acesso de um GroupThis exemplo de EC2 segurança revoga o acesso a um grupo de EC2 segurança da Amazon nomeado. Comando:

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

Revogar o acesso a um CIDR rangeThis exemplo revoga o acesso a um CIDR intervalo. Comando:

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --cidrip 192.168.100.100/32
```

- Para API obter detalhes, consulte [RevokeClusterSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

revoke-snapshot-access

O código de exemplo a seguir mostra como usar `revoke-snapshot-access`.

AWS CLI

Revogar a autorização de uma AWS conta para restaurar um Snapshot. Este exemplo revoga a autorização da AWS conta 444455556666 para restaurar o snapshot. `my-snapshot-id` Por padrão, a saída está no JSON formato. Comando:

```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

Resultado:

```
{
  "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"  
}  
}
```

- Para API obter detalhes, consulte [RevokeSnapshotAccess](#) na Referência de AWS CLI Comandos.

rotate-encryption-key

O código de exemplo a seguir mostra como usar rotate-encryption-key.

AWS CLI

Para rotacionar a chave de criptografia para um cluster

O rotate-encryption-key exemplo a seguir gira a chave de criptografia para o cluster especificado.

```
aws redshift rotate-encryption-key \  
--cluster-identifier mycluster
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Amazon Redshift Database Encryption](#) no Amazon Redshift Cluster Management Guide.

- Para API obter detalhes, consulte [RotateEncryptionKey](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Rekognition usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Rekognition.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

compare-faces

O código de exemplo a seguir mostra como usar `compare-faces`.

Para obter mais informações, consulte [Comparação de faces em imagens](#).

AWS CLI

Como comparar faces em duas imagens

O comando `compare-faces` a seguir compara faces em duas imagens armazenadas em um bucket do Amazon S3.

```
aws rekognition compare-faces \
  --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \
  --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Compare rostos em imagens](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [CompareFaces](#) na Referência de AWS CLI Comandos.

create-collection

O código de exemplo a seguir mostra como usar `create-collection`.

Para obter mais informações, consulte [Criar uma coleção](#).

AWS CLI

Como criar uma coleção

O comando `create-collection` a seguir cria uma coleção com o nome especificado.

```
aws rekognition create-collection \
  --collection-id "MyCollection"
```

Saída:

```
{
  "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
  "FaceModelVersion": "4.0",
  "StatusCode": 200
}
```

Para obter mais informações, consulte [Criar uma coleção](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [CreateCollection](#) na Referência de AWS CLI Comandos.

create-stream-processor

O código de exemplo a seguir mostra como usar `create-stream-processor`.

AWS CLI

Para criar um novo processador de stream

O `create-stream-processor` exemplo a seguir cria um novo processador de stream com a configuração especificada.

```
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}}'
```

Saída:

```
{
  "StreamProcessorArn": "arn:aws:rekognition:us-
west-2:123456789012:streamprocessor/my-stream-processor"
}
```

Para obter mais informações, consulte Como [trabalhar com streaming de vídeos no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [CreateStreamProcessor](#) na Referência de AWS CLI Comandos.

delete-collection

O código de exemplo a seguir mostra como usar `delete-collection`.

Para obter mais informações, consulte [Excluir uma coleção](#).

AWS CLI

Como excluir uma coleção

O comando `delete-collection` a seguir exclui a coleção especificada.

```
aws rekognition delete-collection \  
  --collection-id MyCollection
```

Saída:

```
{  
  "StatusCode": 200  
}
```

Para obter mais informações, consulte [Excluir uma coleção](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DeleteCollection](#) na Referência de AWS CLI Comandos.

delete-faces

O código de exemplo a seguir mostra como usar `delete-faces`.

Para obter mais informações, consulte [Excluir faces de uma coleção](#).

AWS CLI

Como excluir faces de uma coleção

O comando `delete-faces` a seguir exclui uma face especificada de uma coleção.

```
aws rekognition delete-faces \  
  --collection-id MyCollection  
  --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

Saída:

```
{  
  "DeletedFaces": [  
    {  
      "FaceId": "0040279c-0178-436e-b70a-e61b074e96b0"  
    }  
  ]  
}
```

```
    "0040279c-0178-436e-b70a-e61b074e96b0"  
  ]  
}
```

Para obter mais informações, consulte [Excluir faces de uma coleção](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DeleteFaces](#) na Referência de AWS CLI Comandos.

delete-stream-processor

O código de exemplo a seguir mostra como usar `delete-stream-processor`.

AWS CLI

Para excluir um processador de stream

O `delete-stream-processor` comando a seguir exclui o processador de fluxo especificado.

```
aws rekognition delete-stream-processor \  
  --name my-stream-processor
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com streaming de vídeos no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DeleteStreamProcessor](#) na Referência de AWS CLI Comandos.

describe-collection

O código de exemplo a seguir mostra como usar `describe-collection`.

Para obter mais informações, consulte [Descrever uma coleção](#).

AWS CLI

Como descrever uma coleção

O exemplo de `describe-collection` a seguir exibe os detalhes da coleção especificada.

```
aws rekognition describe-collection \  
  --collection-id my-collection-id
```

```
--collection-id MyCollection
```

Saída:

```
{
  "FaceCount": 200,
  "CreationTimestamp": 1569444828.274,
  "CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
  "FaceModelVersion": "4.0"
}
```

Para obter mais informações, consulte [Descrever uma coleção](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DescribeCollection](#) na Referência de AWS CLI Comandos.

describe-stream-processor

O código de exemplo a seguir mostra como usar `describe-stream-processor`.

AWS CLI

Para obter informações sobre um processador de stream

O `describe-stream-processor` comando a seguir exibe detalhes sobre o processador de stream especificado.

```
aws rekognition describe-stream-processor \
  --name my-stream-processor
```

Saída:

```
{
  "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
}

```

Para obter mais informações, consulte Como [trabalhar com streaming de vídeos no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DescribeStreamProcessor](#) na Referência de AWS CLI Comandos.

detect-faces

O código de exemplo a seguir mostra como usar detect-faces.

Para obter mais informações, consulte [Detectar faces em uma imagem](#).

AWS CLI

Como detectar faces em uma imagem

O comando detect-faces a seguir detecta faces na imagem especificada armazenada em um bucket do Amazon S3.

```

aws rekognition detect-faces \
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \
  --attributes "ALL"

```

Saída:

```
{
  "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  
    }  
  }  
]  
}
```

Para obter mais informações, consulte [Detectar faces em uma imagem](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DetectFaces](#) na Referência de AWS CLI Comandos.

detect-labels

O código de exemplo a seguir mostra como usar `detect-labels`.

Para obter mais informações, consulte [Detectar rótulos em uma imagem](#).

AWS CLI

Como detectar rótulos em uma imagem

O exemplo de `detect-labels` a seguir detecta cenas e objetos em uma imagem armazenada em um bucket do Amazon S3.

```
aws rekognition detect-labels \  
  --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Detectar rótulos em uma imagem](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DetectLabels](#) na Referência de AWS CLI Comandos.

detect-moderation-labels

O código de exemplo a seguir mostra como usar `detect-moderation-labels`.

Para obter mais informações, consulte [Detectar imagens impróprias](#).

AWS CLI

Como detectar conteúdo não seguro em uma imagem

O comando `detect-moderation-labels` a seguir detecta conteúdo não seguro na imagem especificada armazenada em um bucket do Amazon S3.

```
aws rekognition detect-moderation-labels \  
  --image "S3Object={Bucket=MyImageS3Bucket, Name=gun.jpg}"
```

Saída:

```
{  
  "ModerationModelVersion": "3.0",  
  "ModerationLabels": [  
    {  
      "Confidence": 97.29618072509766,  
      "ParentName": "Violence",  
      "Name": "Weapon Violence"  
    },  
    {  
      "Confidence": 97.29618072509766,  
      "ParentName": "",  
      "Name": "Violence"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Detectando imagens inapropriadas](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [DetectModerationLabels](#) na Referência de AWS CLI Comandos.

detect-text

O código de exemplo a seguir mostra como usar `detect-text`.

Para obter mais informações, consulte [Detectar texto em uma imagem](#).

AWS CLI

Como detectar texto em uma imagem

O comando `detect-text` a seguir detecta texto na imagem especificada.

```
aws rekognition detect-text \  
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'
```

Saída:

```
{  
  "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
}
```

```
    }  
  ]  
}
```

- Para API obter detalhes, consulte [DetectText](#) na Referência de AWS CLI Comandos.

disassociate-faces

O código de exemplo a seguir mostra como usar `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
```

- Para API obter detalhes, consulte [DisassociateFaces](#) na Referência de AWS CLI Comandos.

get-celebrity-info

O código de exemplo a seguir mostra como usar `get-celebrity-info`.

AWS CLI

Como obter informações sobre uma celebridade

O comando `get-celebrity-info` a seguir exibe informações sobre a celebridade especificada. O parâmetro `id` é proveniente de uma chamada anterior a `recognize-celebrities`.

```
aws rekognition get-celebrity-info --id nnnnnnnn
```

Saída:

```
{  
  "Name": "Celeb A",  
  "Urls": [  
    "www.imdb.com/name/aaaaaaaaa"  
  ]  
}
```

Para obter mais informações, consulte [Obter informações sobre uma celebridade](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [GetCelebrityInfo](#) na Referência de AWS CLI Comandos.

get-celebrity-recognition

O código de exemplo a seguir mostra como usar get-celebrity-recognition.

AWS CLI

Para obter os resultados de uma operação de reconhecimento de celebridades

O get-celebrity-recognition comando a seguir exibe os resultados de uma operação de reconhecimento de celebridades que você iniciou anteriormente ligando. start-celebrity-recognition

```
aws rekognition get-celebrity-recognition \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Saída:

```
{
  "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
}
}

```

Para obter mais informações, consulte Como [reconhecer celebridades em um vídeo armazenado](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [GetCelebrityRecognition](#) na Referência de AWS CLI Comandos.

get-content-moderation

O código de exemplo a seguir mostra como usar `get-content-moderation`.

AWS CLI

Para obter os resultados de uma operação de conteúdo não segura

O `get-content-moderation` comando a seguir exibe os resultados de uma operação de conteúdo não seguro que você iniciou anteriormente `start-content-moderation` chamando.

```
aws rekognition get-content-moderation \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Saída:

```
{
```

```
"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
  }
}
```

Para obter mais informações, consulte [Detecção de vídeos armazenados não seguros](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [GetContentModeration](#) na Referência de AWS CLI Comandos.

get-face-detection

O código de exemplo a seguir mostra como usar `get-face-detection`.

AWS CLI

Para obter os resultados de uma operação de detecção facial

O `get-face-detection` comando a seguir exibe os resultados de uma operação de detecção facial que você iniciou anteriormente por meio de uma chamada `start-face-detection`.

```
aws rekognition get-face-detection \  
--job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Saída:

```
{  
  "Faces": [  
    {  
      "Timestamp": 467,  
      "Face": {  
        "BoundingBox": {  
          "Width": 0.1560753583908081,  
          "Top": 0.13555361330509186,  
          "Left": -0.0952017530798912,  
          "Height": 0.6934483051300049  
        },  
        "Landmarks": [  
          {  
            "Y": 0.4013825058937073,  
            "X": -0.041750285774469376,  
            "Type": "eyeLeft"  
          },  
          {  
            "Y": 0.41695496439933777,  
            "X": 0.027979329228401184,  
            "Type": "eyeRight"  
          },  
          {  
            "Y": 0.6375303268432617,  
            "X": -0.04034662991762161,  
            "Type": "mouthLeft"  
          },  
          {  
            "Y": 0.6497718691825867,  
            "X": 0.013960429467260838,  
            "Type": "mouthRight"  
          }  
        ]  
      }  
    }  
  ]  
}
```



```
    },
    {
      "Y": 0.5238034129142761,
      "X": 0.008022055961191654,
      "Type": "nose"
    }
  ],
  "Pose": {
    "Yaw": -58.07863998413086,
    "Roll": 1.9384294748306274,
    "Pitch": -24.66305160522461
  },
  "Quality": {
    "Sharpness": 83.14741516113281,
    "Brightness": 25.75942611694336
  },
  "Confidence": 87.7622299194336
}
},
{
  "Timestamp": 967,
  "Face": {
    "BoundingBox": {
      "Width": 0.28559377789497375,
      "Top": 0.19436298310756683,
      "Left": 0.024553587660193443,
      "Height": 0.7216082215309143
    },
    "Landmarks": [
      {
        "Y": 0.4650231599807739,
        "X": 0.16269078850746155,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.4843238294124603,
        "X": 0.2782580852508545,
        "Type": "eyeRight"
      },
      {
        "Y": 0.71530681848526,
        "X": 0.1741468608379364,
        "Type": "mouthLeft"
      }
    ]
  }
},
```

```

        {
            "Y": 0.7310671210289001,
            "X": 0.26857468485832214,
            "Type": "mouthRight"
        },
        {
            "Y": 0.582602322101593,
            "X": 0.2566150426864624,
            "Type": "nose"
        }
    ],
    "Pose": {
        "Yaw": 11.487052917480469,
        "Roll": 5.074230670928955,
        "Pitch": 15.396159172058105
    },
    "Quality": {
        "Sharpness": 73.32209777832031,
        "Brightness": 54.96497344970703
    },
    "Confidence": 99.99998474121094
}
}
],
"NextToken":
"OzL223pDKy91160/02KXRqFIEAwxy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
}
}
}

```

Para obter mais informações, consulte [Detecção de rostos em um vídeo armazenado](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [GetFaceDetection](#) na Referência de AWS CLI Comandos.

get-face-search

O código de exemplo a seguir mostra como usar `get-face-search`.

AWS CLI

Para obter os resultados de uma operação de busca facial

O `get-face-search` comando a seguir exibe os resultados de uma operação de busca facial que você iniciou anteriormente chamando `start-face-search`.

```
aws rekognition get-face-search \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Saída:

```
{
  "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+GejyFHPNa/
ldXIC4c/d",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
}
}
}

```

Para obter mais informações, consulte [Pesquisar rostos em vídeos armazenados](#) no Amazon Rekognition Developer Guide.

- Para API obter detalhes, consulte [GetFaceSearch](#) na Referência de AWS CLI Comandos.

get-label-detection

O código de exemplo a seguir mostra como usar `get-label-detection`.

AWS CLI

Para obter os resultados de uma operação de detecção de objetos e cenas

O `get-label-detection` comando a seguir exibe os resultados de uma operação de detecção de objetos e cenas que você iniciou anteriormente chamando `start-label-detection`.

```
aws rekognition get-label-detection \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Saída:

```
{
```

```
"Labels": [
  {
    "Timestamp": 0,
    "Label": {
      "Instances": [],
      "Confidence": 50.19071578979492,
      "Parents": [
        {
          "Name": "Person"
        },
        {
          "Name": "Crowd"
        }
      ],
      "Name": "Audience"
    }
  },
  {
    "Timestamp": 0,
    "Label": {
      "Instances": [],
      "Confidence": 55.74115753173828,
      "Parents": [
        {
          "Name": "Room"
        },
        {
          "Name": "Indoors"
        },
        {
          "Name": "School"
        }
      ],
      "Name": "Classroom"
    }
  }
],
"JobStatus": "SUCCEEDED",
"LabelModelVersion": "2.0",
"VideoMetadata": {
  "Format": "QuickTime / MOV",
  "FrameRate": 29.970617294311523,
  "Codec": "h264",
  "DurationMillis": 6806,
```

```
    "FrameHeight": 1080,  
    "FrameWidth": 1920  
  },  
  "NextToken": "BMugzAi4L72IERzQdbpyMQuEFBsJlo5W0Yx3mfG+sR9mm98E1/  
Cp0benspRfs/5FBQFs4X7G"  
}
```

Para obter mais informações, consulte [Detecção de rótulos em um vídeo no Guia do desenvolvedor do Amazon Rekognition](#).

- Para API obter detalhes, consulte [GetLabelDetection](#) na Referência de AWS CLI Comandos.

get-person-tracking

O código de exemplo a seguir mostra como usar get-person-tracking.

AWS CLI

Para obter os resultados de uma operação de seleção de caminhos de pessoas

O get-person-tracking comando a seguir exibe os resultados de uma operação de definição de caminhos de pessoas que você iniciou anteriormente por meio de uma chamada start-person-tracking.

```
aws rekognition get-person-tracking \  
--job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Saída:

```
{  
  "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": 0.077777778059244156,
          "Left": 0.0,
          "Height": 0.91944444417953491
        },
        "Index": 0
      }
    }
  ],
  "NextToken": "D/vRIYnyhG79ugdta3f+8cRg9oSro
+HigG0uxRiYpTn0ExnqTi1CJektVAc4HrAXDv25eHYk",
  "JobStatus": "SUCCEEDED",
  "VideoMetadata": {
    "Format": "QuickTime / MOV",
    "FrameRate": 29.970617294311523,
    "Codec": "h264",
    "DurationMillis": 6806,
    "FrameHeight": 1080,
    "FrameWidth": 1920
  }
}

```

Para obter mais informações, consulte [People Pathing no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [GetPersonTracking](#) na Referência de AWS CLI Comandos.

index-faces

O código de exemplo a seguir mostra como usar `index-faces`.

Para obter mais informações, consulte [Adicionar faces a uma coleção](#).

AWS CLI

Como adicionar faces a uma coleção

O comando `index-faces` a seguir adiciona as faces encontradas em uma imagem à coleção especificada.

```
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"
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Adicionar faces a uma coleção](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [IndexFaces](#) na Referência de AWS CLI Comandos.

list-collections

O código de exemplo a seguir mostra como usar `list-collections`.

Para obter mais informações, consulte [Listar coleções](#).

AWS CLI

Como listar as coleções disponíveis

O `list-collections` comando a seguir lista as coleções disponíveis na AWS conta.

```
aws rekognition list-collections
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações, consulte [Listar coleções](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [ListCollections](#) na Referência de AWS CLI Comandos.

list-faces

O código de exemplo a seguir mostra como usar `list-faces`.

Para obter mais informações, consulte [Listar faces em uma coleção](#).

AWS CLI

Como listar faces de uma coleção

O comando `list-faces` a seguir lista as faces na coleção especificada.

```
aws rekognition list-faces \
  --collection-id MyCollection
```

Saída:

```
{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Listar faces em uma coleção](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [ListFaces](#) na Referência de AWS CLI Comandos.

list-stream-processors

O código de exemplo a seguir mostra como usar `list-stream-processors`.

AWS CLI

Para listar os processadores de stream em sua conta

O `list-stream-processors` comando a seguir lista os processadores de stream em sua conta e o estado de cada um.

```
aws rekognition list-stream-processors
```

Saída:

```
{
  "StreamProcessors": [
    {
      "Status": "STOPPED",
      "Name": "my-stream-processor"
    }
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com streaming de vídeos no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [ListStreamProcessors](#) na Referência de AWS CLI Comandos.

recognize-celebrities

O código de exemplo a seguir mostra como usar `recognize-celebrities`.

Para obter mais informações, consulte [Reconhecer celebridades em uma imagem](#).

AWS CLI

Como reconhecer celebridades em uma imagem

O seguinte comando `recognize-celebrities` reconhece celebridades na imagem especificada armazenada em um bucket do Amazon S3:

```
aws rekognition recognize-celebrities \
  --image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

Saída:

```
{
  "UnrecognizedFaces": [
    {
```

```
"BoundingBox": {
  "Width": 0.14416666328907013,
  "Top": 0.077777778059244156,
  "Left": 0.625,
  "Height": 0.2746031880378723
},
"Confidence": 99.9990234375,
"Pose": {
  "Yaw": 10.80408763885498,
  "Roll": -12.761146545410156,
  "Pitch": 10.96889877319336
},
"Quality": {
  "Sharpness": 94.1185531616211,
  "Brightness": 79.18367004394531
},
"Landmarks": [
  {
    "Y": 0.18220913410186768,
    "X": 0.6702951788902283,
    "Type": "eyeLeft"
  },
  {
    "Y": 0.16337193548679352,
    "X": 0.7188183665275574,
    "Type": "eyeRight"
  },
  {
    "Y": 0.20739148557186127,
    "X": 0.7055801749229431,
    "Type": "nose"
  },
  {
    "Y": 0.2889308035373688,
    "X": 0.687512218952179,
    "Type": "mouthLeft"
  },
  {
    "Y": 0.2706988751888275,
    "X": 0.7250053286552429,
    "Type": "mouthRight"
  }
]
}
```

```
],
"CelebrityFaces": [
  {
    "MatchConfidence": 100.0,
    "Face": {
      "BoundingBox": {
        "Width": 0.14000000059604645,
        "Top": 0.1190476194024086,
        "Left": 0.82833331823349,
        "Height": 0.2666666805744171
      },
      "Confidence": 99.99359130859375,
      "Pose": {
        "Yaw": -10.509642601013184,
        "Roll": -14.51749324798584,
        "Pitch": 13.799399375915527
      },
      "Quality": {
        "Sharpness": 78.74752044677734,
        "Brightness": 42.201324462890625
      },
      "Landmarks": [
        {
          "Y": 0.2290833294391632,
          "X": 0.8709492087364197,
          "Type": "eyeLeft"
        },
        {
          "Y": 0.20639978349208832,
          "X": 0.9153988361358643,
          "Type": "eyeRight"
        },
        {
          "Y": 0.25417643785476685,
          "X": 0.8907724022865295,
          "Type": "nose"
        },
        {
          "Y": 0.32729196548461914,
          "X": 0.8876466155052185,
          "Type": "mouthLeft"
        },
        {
          "Y": 0.3115464746952057,
```

```
        "X": 0.9238573312759399,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb A",
  "Urls": [
    "www.imdb.com/name/aaaaaaaaa"
  ],
  "Id": "1111111"
},
{
  "MatchConfidence": 97.0,
  "Face": {
    "BoundingBox": {
      "Width": 0.13333334028720856,
      "Top": 0.24920634925365448,
      "Left": 0.4449999928474426,
      "Height": 0.2539682686328888
    },
    "Confidence": 99.99979400634766,
    "Pose": {
      "Yaw": 6.557040691375732,
      "Roll": -7.316643714904785,
      "Pitch": 9.272967338562012
    },
    "Quality": {
      "Sharpness": 83.23492431640625,
      "Brightness": 78.83267974853516
    },
    "Landmarks": [
      {
        "Y": 0.3625510632991791,
        "X": 0.48898839950561523,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.35366007685661316,
        "X": 0.5313721299171448,
        "Type": "eyeRight"
      },
      {
        "Y": 0.3894785940647125,
        "X": 0.5173314809799194,
```

```
        "Type": "nose"
      },
      {
        "Y": 0.44889405369758606,
        "X": 0.5020005702972412,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.4408611059188843,
        "X": 0.5351271629333496,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb B",
  "Urls": [
    "www.imdb.com/name/bbbbbbbbbb"
  ],
  "Id": "2222222"
},
{
  "MatchConfidence": 100.0,
  "Face": {
    "BoundingBox": {
      "Width": 0.12416666746139526,
      "Top": 0.2968254089355469,
      "Left": 0.2150000035762787,
      "Height": 0.23650793731212616
    },
    "Confidence": 99.99958801269531,
    "Pose": {
      "Yaw": 7.801797866821289,
      "Roll": -8.326810836791992,
      "Pitch": 7.844768047332764
    },
    "Quality": {
      "Sharpness": 86.93206024169922,
      "Brightness": 79.81291198730469
    },
    "Landmarks": [
      {
        "Y": 0.4027804136276245,
        "X": 0.2575301229953766,
        "Type": "eyeLeft"
      }
    ]
  }
}
```



```
    },
    {
      "Y": 0.3934555947780609,
      "X": 0.2956969439983368,
      "Type": "eyeRight"
    },
    {
      "Y": 0.4309830069541931,
      "X": 0.2837020754814148,
      "Type": "nose"
    },
    {
      "Y": 0.48186683654785156,
      "X": 0.26812544465065,
      "Type": "mouthLeft"
    },
    {
      "Y": 0.47338807582855225,
      "X": 0.29905644059181213,
      "Type": "mouthRight"
    }
  ]
},
"Name": "Celeb C",
"Urls": [
  "www.imdb.com/name/ccccccccc"
],
"Id": "3333333"
},
{
  "MatchConfidence": 97.0,
  "Face": {
    "BoundingBox": {
      "Width": 0.11916666477918625,
      "Top": 0.3698412775993347,
      "Left": 0.008333333767950535,
      "Height": 0.22698412835597992
    },
    "Confidence": 99.99999237060547,
    "Pose": {
      "Yaw": 16.38478660583496,
      "Roll": -1.0260354280471802,
      "Pitch": 5.975185394287109
    }
  },
}
```

```
    "Quality": {
      "Sharpness": 83.23492431640625,
      "Brightness": 61.408443450927734
    },
    "Landmarks": [
      {
        "Y": 0.4632347822189331,
        "X": 0.049406956881284714,
        "Type": "eyeLeft"
      },
      {
        "Y": 0.46388113498687744,
        "X": 0.08722897619009018,
        "Type": "eyeRight"
      },
      {
        "Y": 0.5020678639411926,
        "X": 0.0758260041475296,
        "Type": "nose"
      },
      {
        "Y": 0.544157862663269,
        "X": 0.054029736667871475,
        "Type": "mouthLeft"
      },
      {
        "Y": 0.5463630557060242,
        "X": 0.08464983850717545,
        "Type": "mouthRight"
      }
    ]
  },
  "Name": "Celeb D",
  "Urls": [
    "www.imdb.com/name/ddddddddd"
  ],
  "Id": "44444444"
}
]
```

Para obter mais informações, consulte [Reconhecer celebridades em uma imagem](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [RecognizeCelebrities](#) na Referência de AWS CLI Comandos.

search-faces-by-image

O código de exemplo a seguir mostra como usar `search-faces-by-image`.

Para obter mais informações, consulte [Pesquisar uma face \(imagem\)](#).

AWS CLI

Como pesquisar faces em uma coleção que corresponda à maior face em uma imagem.

O seguinte comando `search-faces-by-image` pesquisa faces em uma coleção que corresponda à maior face na imagem especificada:

```
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"
}

```

Para obter mais informações, consulte [Procurando um rosto com uma imagem](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [SearchFacesByImage](#) na Referência de AWS CLI Comandos.

search-faces

O código de exemplo a seguir mostra como usar `search-faces`.

Para obter mais informações, consulte [Pesquisar uma face \(face ID\)](#).

AWS CLI

Como pesquisar faces em uma coleção que corresponda a um ID facial.

O comando `search-faces` a seguir pesquisa faces em uma coleção que correspondem ao ID facial especificado.

```
aws rekognition search-faces \  
  --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \  
  --collection-id MyCollection
```

Saída:

```
{  
  "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
}
]
}

```

Para obter mais informações, consulte [Procurando uma face com um ID facial](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [SearchFaces](#) na Referência de AWS CLI Comandos.

start-celebrity-recognition

O código de exemplo a seguir mostra como usar `start-celebrity-recognition`.

AWS CLI

Para iniciar o reconhecimento de celebridades em um vídeo armazenado

O `start-celebrity-recognition` comando a seguir inicia um trabalho para procurar celebridades no arquivo de vídeo especificado armazenado em um bucket do Amazon S3.

```
aws rekognition start-celebrity-recognition \
  --video "S3object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Saída:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Para obter mais informações, consulte Como [reconhecer celebridades em um vídeo armazenado](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StartCelebrityRecognition](#) na Referência de AWS CLI Comandos.

start-content-moderation

O código de exemplo a seguir mostra como usar `start-content-moderation`.

AWS CLI

Para iniciar o reconhecimento de conteúdo não seguro em um vídeo armazenado

O `start-content-moderation` comando a seguir inicia um trabalho para detectar conteúdo não seguro no arquivo de vídeo especificado armazenado em um bucket do Amazon S3.

```
aws rekognition start-content-moderation \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Saída:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Para obter mais informações, consulte [Detecção de vídeos armazenados não seguros](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StartContentModeration](#) na Referência de AWS CLI Comandos.

start-face-detection

O código de exemplo a seguir mostra como usar `start-face-detection`.

AWS CLI

Para detectar rostos em um vídeo

O `start-face-detection` comando a seguir inicia um trabalho para detectar faces no arquivo de vídeo especificado armazenado em um bucket do Amazon S3.

```
aws rekognition start-face-detection
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Saída:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Para obter mais informações, consulte [Detecção de rostos em um vídeo armazenado](#) no Guia do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StartFaceDetection](#) na Referência de AWS CLI Comandos.

start-face-search

O código de exemplo a seguir mostra como usar start-face-search.

AWS CLI

Para pesquisar rostos em uma coleção que correspondam aos rostos detectados em um vídeo

O start-face-search comando a seguir inicia um trabalho para pesquisar faces em uma coleção que correspondam às faces detectadas no arquivo de vídeo especificado em um bucket do Amazon S3.

```
aws rekognition start-face-search \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
  --collection-id collection
```

Saída:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Para obter mais informações, consulte [Pesquisar rostos em vídeos armazenados](#) no Amazon Rekognition Developer Guide.

- Para API obter detalhes, consulte [StartFaceSearch](#) na Referência de AWS CLI Comandos.

start-label-detection

O código de exemplo a seguir mostra como usar `start-label-detection`.

AWS CLI

Para detectar objetos e cenas em um vídeo

O `start-label-detection` comando a seguir inicia um trabalho para detectar objetos e cenas no arquivo de vídeo especificado armazenado em um bucket do Amazon S3.

```
aws rekognition start-label-detection \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Saída:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Para obter mais informações, consulte [Detecção de rótulos em um vídeo no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StartLabelDetection](#) na Referência de AWS CLI Comandos.

start-person-tracking

O código de exemplo a seguir mostra como usar `start-person-tracking`.

AWS CLI

Para iniciar a trajetória das pessoas em um vídeo armazenado

O `start-person-tracking` comando a seguir inicia um trabalho para rastrear os caminhos que as pessoas percorrem no arquivo de vídeo especificado armazenado em um bucket do Amazon S3. :

```
aws rekognition start-person-tracking \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Saída:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Para obter mais informações, consulte [People Pathing no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StartPersonTracking](#) na Referência de AWS CLI Comandos.

start-stream-processor

O código de exemplo a seguir mostra como usar `start-stream-processor`.

AWS CLI

Para iniciar um processador de stream

O `start-stream-processor` comando a seguir inicia o processador de fluxo de vídeo especificado.

```
aws rekognition start-stream-processor \
  --name my-stream-processor
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com streaming de vídeos no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StartStreamProcessor](#) na Referência de AWS CLI Comandos.

stop-stream-processor

O código de exemplo a seguir mostra como usar `stop-stream-processor`.

AWS CLI

Para interromper a execução de um processador de stream

O `stop-stream-processor` comando a seguir interrompe o processador de fluxo em execução especificado.

```
aws rekognition stop-stream-processor \  
  --name my-stream-processor
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com streaming de vídeos no Guia](#) do desenvolvedor do Amazon Rekognition.

- Para API obter detalhes, consulte [StopStreamProcessor](#) na Referência de AWS CLI Comandos.

AWS RAM exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS RAM.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-resource-share-invitation

O código de exemplo a seguir mostra como usar `accept-resource-share-invitation`.

AWS CLI

Para aceitar um convite de compartilhamento de recursos

O `accept-resource-share-invitation` exemplo a seguir aceita o convite de compartilhamento de recursos especificado. Os diretores da conta convidada podem começar imediatamente a usar os recursos do compartilhamento.

```
aws ram accept-resource-share-invitation \  
  --resource-arn arn:aws:ram:us-east-1:123456789012:resource-share/123456789012-123456789012-123456789012
```

```
--resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [AcceptResourceShareInvitation](#) na Referência de AWS CLI Comandos.

associate-resource-share-permission

O código de exemplo a seguir mostra como usar `associate-resource-share-permission`.

AWS CLI

Para associar uma permissão RAM gerenciada a um compartilhamento de recursos

O `associate-resource-share-permission` exemplo a seguir substitui a permissão gerenciada existente para o tipo de recurso relevante pela permissão gerenciada especificada. O acesso a todos os recursos do tipo de recurso relevante é regido pela nova permissão.

```
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
```

Saída:


```
{
  "returnValue": true
}
```

- Para API obter detalhes, consulte [AssociateResourceSharePermission](#) na Referência de AWS CLI Comandos.

associate-resource-share

O código de exemplo a seguir mostra como usar `associate-resource-share`.

AWS CLI

Exemplo 1: Para associar um recurso a um compartilhamento de recursos

O `associate-resource-share` exemplo a seguir adiciona uma configuração de licença ao compartilhamento de recursos especificado.

```
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
```

Saída:

```
{
  "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
    }
  ]
}
```

Exemplo 2: Para associar um diretor a um compartilhamento de recursos

O `associate-resource-share` exemplo a seguir concede acesso ao compartilhamento de recursos especificado a todas as contas na unidade organizacional especificada.

```
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
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [AssociateResourceShare](#) na Referência de AWS CLI Comandos.

create-resource-share

O código de exemplo a seguir mostra como usar `create-resource-share`.

AWS CLI

Exemplo 1: Para criar um compartilhamento de recursos

O `create-resource-share` exemplo a seguir cria um compartilhamento de recursos vazio com o nome especificado. Você deve adicionar separadamente recursos, diretores e permissões ao compartilhamento.

```
aws ram create-resource-share \
```

```
--name MyNewResourceShare
```

Saída:

```
{
  "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
  }
}
```

Exemplo 2: Para criar um compartilhamento de recursos com AWS contas como principais

O `create-resource-share` exemplo a seguir cria um compartilhamento de recursos e concede acesso à AWS conta especificada (222222222222). Se os diretores especificados não fizerem parte da mesma AWS organização, os convites serão enviados e deverão ser aceitos antes que o acesso seja concedido.

```
aws ram create-resource-share \
  --name MyNewResourceShare \
  --principals 222222222222
```

Exemplo 3: Para criar um compartilhamento de recursos restrito à sua AWS organização

O `create-resource-share` exemplo a seguir cria um compartilhamento de recursos restrito às contas da AWS organização da qual sua conta é membro e adiciona a OU especificada como principal. Todas as contas nessa OU podem usar os recursos no compartilhamento de recursos.

```
aws ram create-resource-share \
  --name MyNewResourceShare \
  --no-allow-external-principals \
  --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [CreateResourceShare](#) na Referência de AWS CLI Comandos.

delete-resource-share

O código de exemplo a seguir mostra como usar delete-resource-share.

AWS CLI

Para excluir um compartilhamento de recursos

O delete-resource-share exemplo a seguir exclui o compartilhamento de recursos especificado.

```
aws ram delete-resource-share \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE
```

A saída a seguir indica sucesso:

```
{
  "returnValue": true
}
```

- Para API obter detalhes, consulte [DeleteResourceShare](#) na Referência de AWS CLI Comandos.

disassociate-resource-share-permission

O código de exemplo a seguir mostra como usar disassociate-resource-share-permission.

AWS CLI

Para remover uma permissão RAM gerenciada para um tipo de recurso de um compartilhamento de recursos

O `disassociate-resource-share-permission` exemplo a seguir remove a permissão RAM gerenciada para bancos de dados Glue do compartilhamento de recursos especificado.

```
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
```

Saída:

```
{
  "returnValue": true
}
```

- Para API obter detalhes, consulte [DisassociateResourceSharePermission](#) na Referência de AWS CLI Comandos.

`disassociate-resource-share`

O código de exemplo a seguir mostra como usar `disassociate-resource-share`.

AWS CLI

Para remover um recurso de um compartilhamento de recursos

O `disassociate-resource-share` exemplo a seguir remove o recurso especificado, nesse caso uma VPC sub-rede, do compartilhamento de recursos especificado. Qualquer entidade principal com acesso ao compartilhamento de recursos não pode mais realizar operações nesse recurso.

```
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
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DisassociateResourceShare](#) na Referência de AWS CLI Comandos.

enable-sharing-with-aws-organization

O código de exemplo a seguir mostra como usar `enable-sharing-with-aws-organization`.

AWS CLI

Para permitir o compartilhamento de recursos entre AWS Organizations

O `enable-sharing-with-aws-organization` exemplo a seguir permite o compartilhamento de recursos em sua organização e unidades organizacionais.

```
aws ram enable-sharing-with-aws-organization
```

A seguinte saída indica sucesso.

```
{
  "returnValue": true
}
```

- Para API obter detalhes, consulte [EnableSharingWithAwsOrganization](#) na Referência de AWS CLI Comandos.

get-permission

O código de exemplo a seguir mostra como usar `get-permission`.

AWS CLI

Para recuperar os detalhes de uma permissão RAM gerenciada

O `get-permission` exemplo a seguir exibe os detalhes da versão padrão da permissão RAM gerenciada especificada.

```
aws ram get-permission \  
  --permission-arn arn:aws:ram::aws:permission/  
AWSRAMPermissionGlueTableReadWriteForDatabase
```

Saída:

```
{  
  "permission": {  
    "arn": "arn:aws:ram::aws:permission/  
AWSRAMPermissionGlueTableReadWriteForDatabase",  
    "version": "2",  
    "defaultVersion": true,  
    "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",  
    "resourceType": "glue:Database",  
    "permission": "[\n\"Effect\":\n\"Allow\",  
\"Action\":[\n\"glue:GetTable\n\", \n\"glue:UpdateTable\n\", \n\"glue>DeleteTable\n\", \n\"glue:BatchDeleteTable\n\", \n\"glue:BatchDeleteTableVersion\n\", \n\"glue:GetTableVersion\n\", \n\"glue:GetTableVersions\n\", \n\"glue:GetPartition\n\", \n\"glue:GetPartitions\n\", \n\"glue:BatchGetPartition\n\", \n\"glue:BatchCreatePartition\n\", \n\"glue>CreatePartition\n\", \n\"glue:UpdatePartition\n\", \n\"glue:BatchDeletePartition\n\", \n\"glue>DeletePartition\n\", \n\"glue:GetTables\n\", \n\"glue:SearchTables\n\"]]",  
    "creationTime": 1624912434.431,  
    "lastUpdatedTime": 1624912434.431,  
    "isResourceTypeDefault": false  
  }  
}
```

- Para API obter detalhes, consulte [GetPermission](#) na Referência de AWS CLI Comandos.

get-resource-policies

O código de exemplo a seguir mostra como usar `get-resource-policies`.

AWS CLI

Para obter as políticas de um recurso

O `get-resource-policies` exemplo a seguir exibe as políticas de permissão baseadas em recursos para o recurso especificado associado a um compartilhamento de recursos.

```
aws ram get-resource-policies \  
  --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/  
subnet-0250c25a1fEXAMPLE
```

Saída:

```
{  
  "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"}]}]  
}
```

- Para API obter detalhes, consulte [GetResourcePolicies](#) na Referência de AWS CLI Comandos.

get-resource-share-associations

O código de exemplo a seguir mostra como usar `get-resource-share-associations`.

AWS CLI

Exemplo 1: Para listar todas as associações de recursos para todos os tipos de recursos

O `get-resource-share-associations` exemplo a seguir lista as associações de recursos para todos os tipos de recursos em todos os seus compartilhamentos de recursos.

```
aws ram get-resource-share-associations \  
  --association-type RESOURCE
```


Saída:

```
{
  "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
    }
  ]
}
```

Exemplo 2: Para listar associações principais para um compartilhamento de recursos

O `get-resource-share-associations` exemplo a seguir lista somente as associações principais somente para o compartilhamento de recursos especificado.

```
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
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [GetResourceShareAssociations](#) na Referência de AWS CLI Comandos.

get-resource-share-invitations

O código de exemplo a seguir mostra como usar `get-resource-share-invitations`.

AWS CLI

Para listar seus convites de compartilhamento de recursos

O `get-resource-share-invitations` exemplo a seguir lista seus convites atuais de compartilhamento de recursos.

```
aws ram get-resource-share-invitations
```

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [GetResourceShareInvitations](#) na Referência de AWS CLI Comandos.

get-resource-shares

O código de exemplo a seguir mostra como usar `get-resource-shares`.

AWS CLI

Exemplo 1: Para listar compartilhamentos de recursos que você possui e compartilha com outras pessoas

O `get-resource-shares` exemplo a seguir lista os compartilhamentos de recursos que foram criados e estão sendo compartilhados com outras pessoas.

```
aws ram get-resource-shares \
  --resource-owner SELF
```

Saída:

```

{
  "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
}
]
}

```

Exemplo 2: Para listar compartilhamentos de recursos pertencentes a outras pessoas e compartilhados com você

O `get-resource-shares` exemplo a seguir lista os compartilhamentos de recursos que outras pessoas criaram e compartilharam com você. Neste exemplo, não há nenhum.

```
aws ram get-resource-shares \
  --resource-owner OTHER-ACCOUNTS
```

Saída:

```
{
  "resourceShares": []
}
```

- Para API obter detalhes, consulte [GetResourceShares](#) na Referência de AWS CLI Comandos.

list-pending-invitation-resources

O código de exemplo a seguir mostra como usar `list-pending-invitation-resources`.

AWS CLI

Para listar os recursos que estão disponíveis em um compartilhamento de recursos pendente

O `list-pending-invitation-resources` exemplo a seguir lista todos os recursos que estão no compartilhamento de recursos associado ao convite especificado.

```
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
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListPendingInvitationResources](#) na Referência de AWS CLI Comandos.

list-permissions

O código de exemplo a seguir mostra como usar `list-permissions`.

AWS CLI

Para listar as permissões RAM gerenciadas disponíveis

O `list-permissions` exemplo a seguir lista todas as permissões RAM gerenciadas disponíveis somente para o tipo de recurso de banco de dados AWS Glue.

```
aws ram list-permissions \  
  --resource-type glue:Database
```

Saída:

```
{  
  "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
    }
]
}

```

O `list-permissions` exemplo a seguir exibe as permissões RAM gerenciadas disponíveis para todos os tipos de recursos.

```
aws ram list-permissions
```

Saída:

```

{
  "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
    }
  ]
}

```

- Para API obter detalhes, consulte [ListPermissions](#) na Referência de AWS CLI Comandos.

list-principals

O código de exemplo a seguir mostra como usar `list-principals`.

AWS CLI

Para listar diretores com acesso a um recurso

O `list-principals` exemplo a seguir exibe uma lista dos principais que podem acessar recursos do tipo especificado por meio de qualquer compartilhamento de recursos.

```

aws ram list-principals \
  --resource-type ec2:Subnet

```

Saída:


```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListPrincipals](#) na Referência de AWS CLI Comandos.

list-resource-share-permissions

O código de exemplo a seguir mostra como usar `list-resource-share-permissions`.

AWS CLI

Para listar todas as permissões RAM gerenciadas atualmente anexadas a um compartilhamento de recursos

O `list-resource-share-permissions` exemplo a seguir lista todas as permissões RAM gerenciadas que estão anexadas ao compartilhamento de recursos especificado.

```
aws ram list-resource-share-permissions \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

Saída:

```
{
  "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
    }
]
}

```

- Para API obter detalhes, consulte [ListResourceSharePermissions](#) na Referência de AWS CLI Comandos.

list-resource-types

O código de exemplo a seguir mostra como usar `list-resource-types`.

AWS CLI

Para listar os tipos de recursos que são suportados pelo AWS RAM

O `list-resource-types` exemplo a seguir lista todos os tipos de recursos atualmente suportados pelo AWS RAM.

```
aws ram list-resource-types
```

Saída:

```

{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListResourceTypes](#) na Referência de AWS CLI Comandos.

list-resources

O código de exemplo a seguir mostra como usar `list-resources`.

AWS CLI

Para listar os recursos associados a um compartilhamento de recursos

O `list-resources` exemplo a seguir lista todos os recursos no compartilhamento de recursos especificado que são do tipo de recurso especificado.

```

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

```

Saída:

```

{
  "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
    }
  ]
}

```

```

    }
  ]
}

```

- Para API obter detalhes, consulte [ListResources](#) na Referência de AWS CLI Comandos.

promote-resource-share-created-from-policy

O código de exemplo a seguir mostra como usar `promote-resource-share-created-from-policy`.

AWS CLI

Para promover um compartilhamento de recursos baseado em políticas de recursos até a funcionalidade completa no AWS RAM

O `promote-resource-share-created-from-policy` exemplo a seguir usa um compartilhamento de recursos que você criou implicitamente ao anexar uma política baseada em recursos e o converte em totalmente funcional com o AWS RAM console e suas operações. 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

```

Saída:

```

{
  "returnValue": true
}

```

- Para API obter detalhes, consulte [PromoteResourceShareCreatedFromPolicy](#) na Referência de AWS CLI Comandos.

reject-resource-share-invitation

O código de exemplo a seguir mostra como usar `reject-resource-share-invitation`.

AWS CLI

Para rejeitar um convite de compartilhamento de recursos

O `reject-resource-share-invitation` exemplo a seguir rejeita o convite de compartilhamento de recursos especificado.

```
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
```

Saída:

```
"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"
  }
]
```

- Para API obter detalhes, consulte [RejectResourceShareInvitation](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um compartilhamento de recursos

O `tag-resource` exemplo a seguir adiciona uma chave de tag `project` e um valor associado `lima` ao compartilhamento de recursos especificado.

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um compartilhamento de recursos

O `untag-resource` exemplo a seguir remove a chave da `project` tag e o valor associado do compartilhamento de recursos especificado.

```
aws ram untag-resource \  
  --tag-keys project \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-resource-share

O código de exemplo a seguir mostra como usar `update-resource-share`.

AWS CLI

Para atualizar um compartilhamento de recursos

O `update-resource-share` exemplo a seguir altera o compartilhamento de recursos especificado para permitir diretores externos que não estejam em uma AWS organização.

```
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
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [UpdateResourceShare](#) na Referência de AWS CLI Comandos.

Exemplos do Resource Explorer usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Resource Explorer.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-default-view

O código de exemplo a seguir mostra como usar `associate-default-view`.

AWS CLI

Para definir uma visualização do Resource Explorer como padrão para sua AWS região

O `associate-default-view` exemplo a seguir define uma visualização, conforme especificado por `itsARN`, como a visualização padrão para a AWS região na qual você chama a operação.

```
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
```

Saída:

```
{  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

Para obter mais informações, consulte [Configurando uma exibição padrão em uma AWS região](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [AssociateDefaultView](#) na Referência de AWS CLI Comandos.

batch-get-view

O código de exemplo a seguir mostra como usar `batch-get-view`.

AWS CLI

Para recuperar detalhes sobre várias visualizações do Resource Explorer

O `batch-get-view` exemplo a seguir exibe os detalhes sobre duas visualizações especificadas por suas ARNs. Use espaços para separar o múltiplo ARNs no parâmetro `--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
```

Saída:

```
{
```



```

"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": []
}

```

Para obter mais informações sobre exibições, consulte [Sobre as exibições do Resource Explorer](#) no Guia do AWS Usuário do Resource Explorer.

- Para API obter detalhes, consulte [BatchGetView](#) na Referência de AWS CLI Comandos.

create-index

O código de exemplo a seguir mostra como usar create-index.

AWS CLI

Para ativar o Resource Explorer em uma AWS região criando um índice

O `create-index` exemplo a seguir cria um índice local na AWS região em que a operação é chamada. O gera AWS CLI automaticamente um valor de `client-token` parâmetro aleatório e o inclui na chamada AWS se você não especificar um valor.

```
aws resource-explorer-2 create-index \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
}
```

Depois de criar um índice local, você pode convertê-lo no índice agregador da conta executando o [update-index-type](#) comando.

Para obter mais informações, consulte Como [ativar o Resource Explorer em uma AWS região para indexar seus recursos](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [CreateIndex](#) na Referência de AWS CLI Comandos.

create-view

O código de exemplo a seguir mostra como usar `create-view`.

AWS CLI

Exemplo 1: Para criar uma exibição não filtrada para o índice em uma região AWS

O `create-view` exemplo a seguir cria uma visualização na AWS região especificada que retorna todos os resultados na região sem qualquer filtragem. A exibição inclui o campo opcional `Tags` nos resultados retornados. Como essa visualização é criada na região que contém o índice agregador, ela pode incluir resultados de todas as regiões na conta que contém um índice do Resource Explorer.

```
aws resource-explorer-2 create-view \  
  --view-name My-Main-View \  
  --included-properties Name=tags \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
  }  
}
```

Exemplo 2: Para criar uma visualização que retorne somente recursos associados à Amazon EC2

O seguinte `create-view` cria uma visualização na AWS Região `us-east-1` que retorna somente os recursos na Região que estão associados ao EC2 serviço da Amazon. A exibição inclui o `Tags` campo opcional nos resultados retornados. Como essa visualização é criada na região que contém o índice agregador, ela pode incluir resultados de todas as regiões na conta que contêm um índice do 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
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Criação de exibições para pesquisa](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [CreateView](#) na Referência de AWS CLI Comandos.

delete-index

O código de exemplo a seguir mostra como usar `delete-index`.

AWS CLI

Para desativar o Resource Explorer em uma AWS região excluindo seu índice

O `delete-index` exemplo a seguir exclui o índice especificado do Resource Explorer na AWS região em que você faz a solicitação.

```
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
```

Saída:

```
{
```

```
"Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
  "State": "DELETING"
}
```

Para obter mais informações sobre como excluir um índice, consulte Como [desativar o AWS Resource Explorer em uma AWS região](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [DeleteIndex](#) na Referência de AWS CLI Comandos.

delete-view

O código de exemplo a seguir mostra como usar delete-view.

AWS CLI

Para excluir uma exibição do Resource Explorer

O delete-view exemplo a seguir exclui uma visualização especificada por 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
```

Saída:

```
{
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Excluindo exibições](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [DeleteView](#) na Referência de AWS CLI Comandos.

disassociate-default-view

O código de exemplo a seguir mostra como usar disassociate-default-view.

AWS CLI

Para remover a visualização padrão do Resource Explorer de uma AWS região

O seguinte `disassociate-default-view` remove a visualização padrão do Resource Explorer para a AWS região na qual você chama a operação. Depois de realizar essa operação, todas as operações de pesquisa na região devem especificar explicitamente uma visualização ou a operação falhará.

```
aws resource-explorer-2 disassociate-default-view
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando uma exibição padrão em uma AWS região](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [DisassociateDefaultView](#) na Referência de AWS CLI Comandos.

get-default-view

O código de exemplo a seguir mostra como usar `get-default-view`.

AWS CLI

Para recuperar a visualização do Resource Explorer, que é a visualização padrão para sua região AWS

O `get-default-view` exemplo a seguir recupera ARN a visualização que é padrão para a AWS região na qual você chama a operação.

```
aws resource-explorer-2 get-default-view
```

Saída:

```
{
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Configurando uma exibição padrão em uma AWS região](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [GetDefaultView](#) na Referência de AWS CLI Comandos.

get-index

O código de exemplo a seguir mostra como usar `get-index`.

AWS CLI

Exemplo 1: Para recuperar os detalhes de um índice agregador do Resource Explorer

O `get-index` exemplo a seguir exibe os detalhes do índice do Resource Explorer na AWS região especificada. Como a região especificada contém o índice agregador da conta, a saída lista as regiões que replicam dados no índice dessa região.

```
aws resource-explorer-2 get-index \  
  --region us-east-1
```

Saída:

```
{  
  "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"  
}
```

Exemplo 2: Para recuperar os detalhes de um índice local do Resource Explorer

O `get-index` exemplo a seguir exibe os detalhes do índice do Resource Explorer na AWS região especificada. Como a região especificada contém um índice local, a saída lista a região para a qual ela replica os dados do índice dessa região.

```
aws resource-explorer-2 get-index \  
  --region us-west-2
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre índices, consulte [Verificando quais AWS regiões têm o Resource Explorer ativado no](#) Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [GetIndex](#) na Referência de AWS CLI Comandos.

get-view

O código de exemplo a seguir mostra como usar `get-view`.

AWS CLI

Para recuperar detalhes sobre uma exibição do Resource Explorer

O `get-view` exemplo a seguir exibe os detalhes sobre uma visualização especificada por 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
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações sobre exibições, consulte [Sobre as exibições do Resource Explorer](#) no Guia do AWS Usuário do Resource Explorer.

- Para API obter detalhes, consulte [GetView](#) na Referência de AWS CLI Comandos.

list-indexes

O código de exemplo a seguir mostra como usar `list-indexes`.

AWS CLI

Para listar as AWS regiões nas quais o Resource Explorer tem índices

O `list-indexes` exemplo a seguir lista os índices de todas as regiões em que o Resource Explorer tem um índice. A resposta especifica o tipo de cada índice, sua AWS região e seu ARN.

```
aws resource-explorer-2 list-indexes
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações sobre índices, consulte [Verificando quais AWS regiões têm o Resource Explorer ativado no](#) Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [ListIndexes](#) na Referência de AWS CLI Comandos.

list-supported-resource-types

O código de exemplo a seguir mostra como usar `list-supported-resource-types`.

AWS CLI

Para listar as AWS regiões nas quais o Resource Explorer tem índices

O `list-supported-resource-types` exemplo a seguir lista todos os tipos de recursos atualmente suportados pelo &AREXlong;. O exemplo de resposta inclui um `NextToken` valor que indica que há mais saída disponível para recuperação com chamadas adicionais.

```
aws resource-explorer-2 list-supported-resource-types \
  --max-items 10
```

Saída:

```
{
  "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="  
}
```

Para obter a próxima parte da saída, chame a operação novamente e transmita o valor de NextToken resposta da chamada anterior como valor para `--starting-token`. Repita até que NextToken esteja ausente da resposta.

```
aws resource-explorer-2 list-supported-resource-types \  
  --max-items 10 \  
  --starting-  
token eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=
```

Saída:

```
{  
  "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="
}

```

Para obter mais informações sobre índices, consulte [Verificando quais AWS regiões têm o Resource Explorer ativado no](#) Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [ListSupportedResourceTypes](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags anexadas a uma exibição ou índice do Resource Explorer

O `list-tags-for-resource` exemplo a seguir lista os pares de tag, chave e valor anexados para exibição com o especificado ARN. Você deve chamar a operação da AWS região que contém o recurso.

```

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

```

Saída:

```
{
  "Tags": {
    "application": "MainCorpApp",
    "department": "1234"
  }
}
```

Para obter mais informações sobre a marcação de visualizações, consulte Como [marcar visualizações para controle de acesso](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-views

O código de exemplo a seguir mostra como usar `list-views`.

AWS CLI

Para listar as visualizações do Resource Explorer disponíveis em uma AWS região

O `list-views` exemplo a seguir lista todas as visualizações disponíveis na região na qual você invoca a operação.

```
aws resource-explorer-2 list-views
```

Saída:

```
{
  "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"
  ]
}
```

Para obter mais informações sobre exibições, consulte [Sobre as exibições do Resource Explorer](#) no Guia do AWS Usuário do Resource Explorer.

- Para API obter detalhes, consulte [ListViews](#) na Referência de AWS CLI Comandos.

search

O código de exemplo a seguir mostra como usar `search`.

AWS CLI

Exemplo 1: Para pesquisar usando a exibição padrão

O `search` exemplo a seguir exibe todos os recursos especificados que estão associados ao serviço. A pesquisa usa a visualização padrão para a região. O exemplo de resposta inclui um `NextToken` valor que indica que há mais saída disponível para recuperação com chamadas adicionais.

```
aws resource-explorer-2 search \  
  --query-string "service:iam"
```

Saída:

```
{  
  "Count": {  
    "Complete": true,  
    "TotalResources": 55  
  },  
  "NextToken":  
  "AG9V0EF1KLEXAMPLE0hJHVwo5chEXAMPLER5XiEpNrgsEXAMPLE...b0Cm0F0ryHEXAMPLE",  
  "Resources": [{  
    "Arn": "arn:aws:iam::123456789012:policy/service-role/Some-Policy-For-A-  
Service-Role",  
    "LastReportedAt": "2022-07-21T12:34:42Z",  
    "OwningAccountId": "123456789012",  
    "Properties": [],  
    "Region": "global",  
    "ResourceType": "iam:policy",  
    "Service": "iam"  
  }, {  
    "Arn": "arn:aws:iam::123456789012:policy/service-role/Another-Policy-For-A-  
Service-Role",  
    "LastReportedAt": "2022-07-21T12:34:42Z",  
    "OwningAccountId": "123456789012",  
    "Properties": [],  
    "Region": "global",  
    "ResourceType": "iam:policy",  
    "Service": "iam"
```

```

    }, {
      ... TRUNCATED FOR BREVITY ...
    }],
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/my-default-view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }

```

Exemplo 2: Para pesquisar usando uma visualização especificada

O search exemplo de pesquisa a seguir exibe todos os recursos (“*”) na AWS região especificada que são visíveis por meio da visualização especificada. Os resultados incluem somente recursos associados à Amazon EC2 por causa dos filtros anexados à visualização.

```

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

```

Saída:

```

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"  
}
```

Para obter mais informações, consulte [Usando o AWS Resource Explorer para pesquisar recursos](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [Pesquisar](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para marcar uma exibição do Resource Explorer

O `tag-resource` exemplo a seguir adiciona a chave de tag “ambiente” com o valor “produção” à exibição com o especificadoARN.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar visualizações para controle de acesso](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de uma visualização do Resource Explorer

O `untag-resource` exemplo a seguir remove qualquer tag com o nome de chave “ambiente” da exibição com o especificadoARN.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar visualizações para controle de acesso](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-index-type

O código de exemplo a seguir mostra como usar `update-index-type`.

AWS CLI

Para alterar o tipo de um índice do Resource Explorer

O `update-index-type` exemplo a seguir converte o índice especificado de tipo `local aggregator` para tipo `aggregator` para ativar a capacidade de pesquisar recursos em todas as AWS regiões da conta. Você deve enviar a solicitação para a AWS região que contém o índice que você deseja atualizar.

```
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
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre como alterar o tipo de um índice, consulte Como [ativar a pesquisa entre regiões criando um índice agregador](#) no Guia do usuário do AWS Resource Explorer.

- Para API obter detalhes, consulte [UpdateIndexType](#) na Referência de AWS CLI Comandos.

update-view

O código de exemplo a seguir mostra como usar update-view.

AWS CLI

Exemplo 1: Para atualizar o IncludedProperties campo para uma exibição do Resource Explorer

O update-view exemplo a seguir atualiza a exibição especificada adicionando `tags` à opcional `IncludedProperties`. Depois de executar essa operação, as operações de pesquisa que usam essa exibição incluem informações sobre as tags anexadas aos recursos que aparecem nos resultados.

```
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
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}
```

Exemplo 2: Para atualizar os filtros anexados a uma exibição

O `update-view` exemplo a seguir atualiza a visualização especificada para usar um filtro que limita os resultados somente aos tipos de recursos associados ao EC2 serviço da Amazon.

```
aws resource-explorer-2 update-view \  
  --filters FilterString="service:ec2" \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações sobre exibições, consulte [Sobre as exibições do Resource Explorer](#) no Guia do AWS Usuário do Resource Explorer.

- Para API obter detalhes, consulte [UpdateView](#) na Referência de AWS CLI Comandos.

Exemplos de Resource Groups usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Resource Groups.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-group

O código de exemplo a seguir mostra como usar `create-group`.

AWS CLI

Exemplo 1: Para criar um grupo de recursos baseado em tags

O `create-group` exemplo a seguir cria um grupo de recursos baseado em tags de EC2 instâncias da Amazon na região atual. É baseado em uma consulta de recursos marcados com a chave `Name` e o valor `WebServers`. O nome do grupo é `tbq-WebServer`. A consulta está em um JSON arquivo separado que é passado para o comando.

```
aws resource-groups create-group \  
  --name tbq-WebServer \  
  --resource-query file://query.json
```

Conteúdo de `query.json`:

```
{  
  "Type": "TAG_FILTERS_1_0",  
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\":  
  [{\"Key\": \"Name\", \"Values\": [\"WebServers\"]}]}"
```

Saída:

```
{  
  "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\"]]}\""}
  }
}

```

Exemplo 2: Para criar um grupo de recursos CloudFormation baseado em pilhas

O `create-group` exemplo a seguir cria um grupo de recursos AWS CloudFormation baseado em pilhas chamado `sampleCFNstackgroup`. A consulta inclui todos os recursos na CloudFormation pilha especificada que são compatíveis com AWS Resource Groups.

```

aws resource-groups create-group \
  --name cbq-CFNstackgroup \
  --resource-query file://query.json

```

Conteúdo de `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\"}"}
}

```

Saída:

```

{
  "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\"}"}
}

```

```
}  
}
```

Para obter mais informações, consulte [Create Groups](#) no AWS Resource Groups User Guide.

- Para API obter detalhes, consulte [CreateGroup](#) na Referência de AWS CLI Comandos.

delete-group

O código de exemplo a seguir mostra como usar `delete-group`.

AWS CLI

Para atualizar a descrição de um grupo de recursos

O `delete-group` exemplo a seguir atualiza o grupo de recursos especificado.

```
aws resource-groups delete-group \  
  --group-name tbq-WebServer
```

Saída:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-  
WebServer",  
    "Name": "tbq-WebServer"  
  }  
}
```

Para obter mais informações, consulte [Delete Groups](#) no AWS Resource Groups User Guide.

- Para API obter detalhes, consulte [DeleteGroup](#) na Referência de AWS CLI Comandos.

get-group-query

O código de exemplo a seguir mostra como usar `get-group-query`.

AWS CLI

Para anexar a consulta a um grupo de recursos

O `get-group-query` exemplo a seguir exibe uma consulta anexada ao grupo de recursos especificado.

```
aws resource-groups get-group-query \  
  --group-name tbq-WebServer
```

Saída:

```
{  
  "GroupQuery": {  
    "GroupName": "tbq-WebServer",  
    "ResourceQuery": {  
      "Type": "TAG_FILTERS_1_0",  
      "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [{\"Key\": \"Name\", \"Values\": [\"WebServers\"]}]}"  
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetGroupQuery](#) na Referência de AWS CLI Comandos.

get-group

O código de exemplo a seguir mostra como usar `get-group`.

AWS CLI

Para obter informações sobre um grupo de recursos

O `get-group` exemplo a seguir exibe detalhes sobre o grupo de recursos especificado. Para anexar a consulta ao grupo, use `get-group-query`.

```
aws resource-groups get-group \  
  --group-name tbq-WebServer
```

Saída:

```
{  
  "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."
  }
}
```

- Para API obter detalhes, consulte [GetGroup](#) na Referência de AWS CLI Comandos.

get-tags

O código de exemplo a seguir mostra como usar `get-tags`.

AWS CLI

Para recuperar as tags anexadas a um grupo de recursos

O `get-tags` exemplo a seguir exibe os pares de tag, chave e valor anexados ao grupo de recursos especificado (o grupo em si, não seus membros).

```
aws resource-groups get-tags \
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

Saída:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
  "Tags": {
    "QueryType": "tags",
    "QueryResources": "ec2-instances"
  }
}
```

- Para API obter detalhes, consulte [GetTags](#) na Referência de AWS CLI Comandos.

list-group-resources

O código de exemplo a seguir mostra como usar `list-group-resources`.

AWS CLI

Para listar todos os recursos em um grupo de recursos

Exemplo 1: O `list-resource-groups` exemplo a seguir lista todos os recursos que fazem parte do grupo de recursos especificado.

```
aws resource-groups list-group-resources \  
  --group-name tbq-WebServer
```

Saída:

```
{  
  "ResourceIdentifiers": [  
    {  
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/  
i-09f77fa38c12345ab",  
      "ResourceType": "AWS::EC2::Instance"  
    }  
  ]  
}
```

Exemplo 2: O exemplo a seguir lista todos os recursos do grupo que também têm um 'tipo de recurso' 'AWS:::Instance' EC2 . :

```
aws resource-groups list-group-resources --group-name tbq- WebServer --filters name=Tipo de  
recurso, Values=::: :Instância AWS EC2
```

- Para API obter detalhes, consulte [ListGroupResources](#) na Referência de AWS CLI Comandos.

list-groups

O código de exemplo a seguir mostra como usar `list-groups`.

AWS CLI

Para listar os grupos de recursos disponíveis

O `list-groups` exemplo a seguir exibe uma lista de todos os grupos de recursos.

```
aws resource-groups list-groups
```

Saída:

```
{  
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListGroups](#) na Referência de AWS CLI Comandos.

list-resource-groups

O código de exemplo a seguir mostra como usar `list-resource-groups`.

AWS CLI

Para listar todos os recursos em um grupo de recursos

O `list-resource-groups` exemplo a seguir lista todos os recursos que fazem parte do grupo de recursos especificado.

```

aws resource-groups list-group-resources \
  --group-name tbq-WebServer

```

Saída:

```
{
  "ResourceIdentifiers": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
      "ResourceType": "AWS::EC2::Instance"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListResourceGroups](#) na Referência de AWS CLI Comandos.

put-group-configuration

O código de exemplo a seguir mostra como usar `put-group-configuration`.

AWS CLI

Para anexar uma configuração de serviço a um grupo de recursos

Exemplo 1: O `put-group-configuration` exemplo a seguir especifica que o grupo de recursos deve conter somente reservas de EC2 capacidade da Amazon para instâncias nas M5 famílias C5 ou.

```
aws resource-groups put-group-configuration \
  --group MyTestGroup \
  --configuration file://config.json
```

Conteúdo de `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" ]
      }
    ]
  }
]

```

Se for bem-sucedido, esse comando não gerará nenhum resultado.

Para obter mais informações, consulte [Configurações de serviço para grupos de recursos](#) no Resource Groups API Reference Guide.

- Para API obter detalhes, consulte [PutGroupConfiguration](#) na Referência de AWS CLI Comandos.

search-resources

O código de exemplo a seguir mostra como usar `search-resources`.

AWS CLI

Para encontrar recursos que correspondam a uma consulta

O `search-resources` exemplo a seguir recupera uma lista de todos os AWS recursos que correspondem à consulta especificada.

```
aws resource-groups search-resources \
  --resource-query file://query.json
```

Conteúdo de `query.json`:

```
{
  "Type": "TAG_FILTERS_1_0",
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [ { \"Key\": \"Patch Group\", \"Values\": [\"Dev\"] } ]}"
}
```

Saída:

```
{
  "ResourceIdentifiers": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-01a23bc45d67890ef",
      "ResourceType": "AWS::EC2::Instance"
    }
  ]
}
```

- Para API obter detalhes, consulte [SearchResources](#) na Referência de AWS CLI Comandos.

tag

O código de exemplo a seguir mostra como usar tag.

AWS CLI

Para anexar uma tag a um grupo de recursos

O tag exemplo a seguir anexa os pares de chave e valor da tag especificados ao grupo de recursos especificado (o grupo em si, não seus membros).

```
aws resource-groups tag \
  --tags QueryType=tags, QueryResources=ec2-instances \
  --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer
```

Saída:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
  "Tags": {
```

```
    "QueryType": "tags",
    "QueryResources": "ec2-instances"
  }
}
```

Para obter mais informações, consulte [Manage tags](#) no AWS Resource Groups User Guide.

- Para API obter detalhes, consulte [Tag](#) in AWS CLI Command Reference.

untag

O código de exemplo a seguir mostra como usar `untag`.

AWS CLI

Para remover tags de um grupo de recursos

O `untags` exemplo a seguir remove qualquer tag com a chave especificada do próprio grupo de recursos, não de seus membros.

```
aws resource-groups untag \
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \
  --keys QueryType
```

Saída:

```
{
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
  "Keys": [
    "QueryType"
  ]
}
```

Para obter mais informações, consulte [Manage tags](#) no AWS Resource Groups User Guide.

- Para API obter detalhes, consulte [Desmarcar](#) na Referência de AWS CLI Comandos.

update-group-query

O código de exemplo a seguir mostra como usar `update-group-query`.

AWS CLI

Exemplo 1: Para atualizar a consulta de um grupo de recursos baseado em tags

O `update-group-query` exemplo a seguir atualiza a consulta anexada ao grupo de recursos baseado em tags especificado.

```
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\\"]}]}"}'
```

Saída:

```
{
  "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\\"]}]}"}
  }
}
```

Para obter mais informações, consulte [Update Groups](#) no AWS Resource Groups User Guide.

Exemplo 2: Para atualizar a consulta de um grupo de recursos CloudFormation baseado em pilhas

O `update-group-query` exemplo a seguir atualiza a consulta anexada ao grupo de recursos AWS CloudFormation baseado em pilha especificado.

```
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\\"}"}'
```

Saída:

```
{
  "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\"}"
  }
}
```

Para obter mais informações, consulte [Update Groups](#) no AWS Resource Groups User Guide.

- Para API obter detalhes, consulte [UpdateGroupQuery](#) na Referência de AWS CLI Comandos.

update-group

O código de exemplo a seguir mostra como usar update-group.

AWS CLI

Para atualizar a descrição de um grupo de recursos

O update-group exemplo a seguir atualiza a descrição do grupo de recursos especificado.

```
aws resource-groups update-group \
  --group-name tbq-WebServer \
  --description "Resource group for all web server resources."
```

Saída:

```
{
  "Group": {
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
    "Name": "tbq-WebServer"
  }
}
```

```
    "Description": "Resource group for all web server resources."
  }
}
```

Para obter mais informações, consulte [Update Groups](#) no AWS Resource Groups User Guide.

- Para API obter detalhes, consulte [UpdateGroup](#) na Referência de AWS CLI Comandos.

API Exemplos de marcação de Resource Groups usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Resource Groups Tagging API.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-resources

O código de exemplo a seguir mostra como usar `get-resources`.

AWS CLI

Para obter uma lista de recursos marcados

O `get-resources` exemplo a seguir exibe uma lista de recursos na conta que estão marcados com o nome e o valor da chave especificados.

```
aws resourcegroupstaggingapi get-resources \
  --tag-filters Key=Environment,Values=Production \
  --tags-per-page 100
```

Saída:

```
{
  "ResourceTagMappingList": [
    {
      "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-7sbz2Kz0",
      "Tags": [
        {
          "Key": "Environment",
          "Value": "Production"
        }
      ]
    }
  ]
}
```

Para obter mais informações, consulte [GetResources](#) na Resource Groups Tagging API Reference.

- Para API obter detalhes, consulte [GetResources](#) na Referência de AWS CLI Comandos.

get-tag-keys

O código de exemplo a seguir mostra como usar `get-tag-keys`.

AWS CLI

Para obter uma lista de todas as chaves de tag

O `get-tag-keys` exemplo a seguir recupera a lista de todos os nomes de chave de tag usados pelos recursos na conta.

```
aws resourcegroupstaggingapi get-tag-keys
```

Saída:

```
{
  "TagKeys": [
    "Environment",
    "CostCenter",
```

```
    "Department"  
  ]  
}
```

Para obter mais informações, consulte [GetTagKeys](#) na Resource Groups Tagging API Reference.

- Para API obter detalhes, consulte [GetTagKeys](#) na Referência de AWS CLI Comandos.

get-tag-values

O código de exemplo a seguir mostra como usar `get-tag-values`.

AWS CLI

Para obter uma lista de todos os valores de tag

O `get-tag-values` exemplo a seguir exibe todos os valores usados para a chave especificada para todos os recursos no

```
aws resourcegroupstaggingapi get-tag-values \  
  --key=Environment
```

Saída:

```
{  
  "TagValues": [  
    "Alpha",  
    "Gamma",  
    "Production"  
  ]  
}
```

Para obter mais informações, consulte [GetTagValues](#) na Resource Groups Tagging API Reference.

- Para API obter detalhes, consulte [GetTagValues](#) na Referência de AWS CLI Comandos.

tag-resources

O código de exemplo a seguir mostra como usar `tag-resources`.

AWS CLI

Para anexar uma tag a um recurso

O `tag-resources` exemplo a seguir marca o recurso especificado com um nome e um valor de chave.

```
aws resourcegroupstaggingapi tag-resources \  
  --resource-arn-list arn:aws:s3:::MyProductionBucket \  
  --tags Environment=Production, CostCenter=1234
```

Saída:

```
{  
  "FailedResourcesMap": {}  
}
```

Para obter mais informações, consulte [TagResources](#) na Resource Groups Tagging API Reference.

- Para API obter detalhes, consulte [TagResources](#) na Referência de AWS CLI Comandos.

untag-resources

O código de exemplo a seguir mostra como usar `untag-resources`.

AWS CLI

Para remover uma tag de um recurso

O `untag-resources` exemplo a seguir remove as chaves de tag especificadas e todos os valores associados do recurso especificado.

```
aws resourcegroupstaggingapi untag-resources \  
  --resource-arn-list arn:aws:s3:::awsexamplebucket \  
  --tag-keys Environment CostCenter
```

Saída:

```
{  
  "FailedResourcesMap": {}  
}
```

Para obter mais informações, consulte [UntagResources](#) na Resource Groups Tagging API Reference.

- Para API obter detalhes, consulte [UntagResources](#) na Referência de AWS CLI Comandos.

AWS RoboMaker exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS RoboMaker.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-describe-simulation-job

O código de exemplo a seguir mostra como usar `batch-describe-simulation-job`.

AWS CLI

Para descrever em lote os trabalhos de simulação

O `batch-describe-simulation-job` exemplo a seguir recupera detalhes dos três trabalhos de simulação especificados.

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
```

Saída:

```
{
  "jobs": [
    {
      "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x",
      "status": "Completed",
      "lastUpdatedAt": 1548959178.0,
      "failureBehavior": "Continue",
      "clientRequestToken": "6020408e-b05c-4310-9f13-4ed71c5221ed",
      "outputLocation": {
        "s3Bucket": "awsrobomakerobjecttracker-111111111-bundlesbucket-2lk584kiq1oa",
        "s3Prefix": "output"
      },
      "maxJobDurationInSeconds": 3600,
      "simulationTimeMillis": 0,
      "iamRole": "arn:aws:iam::111111111111:role/AWSRoboMakerObjectTracker-154895-SimulationJobRole-14D5ASA7PQE3A",
      "simulationApplications": [
        {
          "application": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/AWSRoboMakerObjectTracker-1548959046124_NPvyfcatq/1548959170096",
          "applicationVersion": "$LATEST",
          "launchConfig": {
            "packageName": "object_tracker_simulation",
            "launchFile": "local_training.launch",
            "environmentVariables": {
              "MARKOV_PRESET_FILE": "object_tracker.py",
              "MODEL_S3_BUCKET": "awsrobomakerobjecttracker-111111111-bundlesbucket-2lk584kiq1oa",
              "MODEL_S3_PREFIX": "model-store",
              "ROS_AWS_REGION": "us-west-2"
            }
          }
        }
      ]
    },
    {
      "tags": {},
      "vpcConfig": {
        "subnets": [
          "subnet-716dd52a",
          "subnet-43c22325",

```



```

        "subnet-3f526976"
    ],
    "securityGroups": [
        "sg-3fb40545"
    ],
    "vpcId": "vpc-99895eff",
    "assignPublicIp": true
}
},
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
p0cpdrrwng2n",
    "status": "Completed",
    "lastUpdatedAt": 1548168817.0,
    "failureBehavior": "Continue",
    "clientRequestToken": "e4a23e75-f9a7-411d-835f-21881c82c58b",
    "outputLocation": {
        "s3Bucket": "awsrobomakercloudwatch-111111111111-
bundlesbucket-14e5s9jvwtmv7",
        "s3Prefix": "output"
    },
    "maxJobDurationInSeconds": 3600,
    "simulationTimeMillis": 0,
    "iamRole": "arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G00BWTQ8YBG6",
    "robotApplications": [
        {
            "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerCloudWatch-1547663411642_NZbpqEJ3T/1547663517377",
            "applicationVersion": "$LATEST",
            "launchConfig": {
                "packageName": "cloudwatch_robot",
                "launchFile": "await_commands.launch",
                "environmentVariables": {
                    "LAUNCH_ID": "1548168752173",
                    "ROS_AWS_REGION": "us-west-2"
                }
            }
        }
    ]
},
    "simulationApplications": [
        {

```

```

        "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LI6D1h6/1547663521470",
        "applicationVersion": "$LATEST",
        "launchConfig": {
            "packageName": "cloudwatch_simulation",
            "launchFile": "bookstore_turtlebot_navigation.launch",
            "environmentVariables": {
                "LAUNCH_ID": "1548168752173",
                "ROS_AWS_REGION": "us-west-2",
                "TURTLEBOT3_MODEL": "waffle_pi"
            }
        }
    },
    "tags": {},
    "vpcConfig": {
        "subnets": [
            "subnet-716dd52a",
            "subnet-43c22325",
            "subnet-3f526976"
        ],
        "securityGroups": [
            "sg-3fb40545"
        ],
        "vpcId": "vpc-99895eff",
        "assignPublicIp": true
    }
},
{
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
g8h6tglmblgw",
    "status": "Canceled",
    "lastUpdatedAt": 1546543442.0,
    "failureBehavior": "Fail",
    "clientRequestToken": "d796bbb4-2a2c-1abc-f2a9-0d9e547d853f",
    "outputLocation": {
        "s3Bucket": "sample-bucket",
        "s3Prefix": "SimulationLog_115490482698"
    },
    "maxJobDurationInSeconds": 28800,
    "simulationTimeMillis": 0,
    "iamRole": "arn:aws:iam:111111111111:role/RoboMakerSampleTheFirst",
    "robotApplications": [

```

```

    {
      "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/RoboMakerHelloWorldRobot/1546541208251",
      "applicationVersion": "$LATEST",
      "launchConfig": {
        "packageName": "hello_world_robot",
        "launchFile": "rotate.launch"
      }
    },
    "simulationApplications": [
      {
        "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
RoboMakerHelloWorldSimulation/1546541198985",
        "applicationVersion": "$LATEST",
        "launchConfig": {
          "packageName": "hello_world_simulation",
          "launchFile": "empty_world.launch"
        }
      }
    ],
    "tags": {}
  }
],
"unprocessedJobs": []
}

```

- Para API obter detalhes, consulte [BatchDescribeSimulationJob](#) na Referência de AWS CLI Comandos.

cancel-simulation-job

O código de exemplo a seguir mostra como usar `cancel-simulation-job`.

AWS CLI

Para cancelar um trabalho de simulação

O `cancel-simulation-job` exemplo a seguir cancela o trabalho de simulação especificado.

```
aws robomaker cancel-simulation-job \
```

```
--job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- Para API obter detalhes, consulte [CancelSimulationJob](#) na Referência de AWS CLI Comandos.

create-deployment-job

O código de exemplo a seguir mostra como usar `create-deployment-job`.

AWS CLI

Para criar um trabalho de implantação

Este exemplo cria um trabalho de implantação para a frota MyFleet. Ela inclui uma variável de ambiente chamada "ENVIRONMENT". Ele também anexa uma tag chamada "Região".

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
```

Saída:

```
{
  "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"
}
}
```

- Para API obter detalhes, consulte [CreateDeploymentJob](#) na Referência de AWS CLI Comandos.

create-fleet

O código de exemplo a seguir mostra como usar `create-fleet`.

AWS CLI

Para criar uma frota

Este exemplo cria uma frota. Ele anexa uma tag chamada Região.

Comando:

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

Saída:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyOtherFleet/1550771394395",
  "name": "MyFleet",
  "createdAt": 1550771394.0,
  "tags": {
    "Region": "East"
  }
}
```

- Para API obter detalhes, consulte [CreateFleet](#) na Referência de AWS CLI Comandos.

create-robot-application-version

O código de exemplo a seguir mostra como usar `create-robot-application-version`.

AWS CLI

Para criar uma versão do aplicativo de robô

Este exemplo cria uma versão do aplicativo robótico.

Comando:

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

Saída:

```
{
  "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": {}
}
```

- Para API obter detalhes, consulte [CreateRobotApplicationVersion](#) na Referência de AWS CLI Comandos.

create-robot-application

O código de exemplo a seguir mostra como usar `create-robot-application`.

AWS CLI

Para criar um aplicativo de robô

Este exemplo cria um aplicativo robótico.

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
```

Saída:

```
{  
  "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": {}  
}
```

- Para API obter detalhes, consulte [CreateRobotApplication](#) na Referência de AWS CLI Comandos.

create-robot

O código de exemplo a seguir mostra como usar `create-robot`.

AWS CLI

Para criar um robô

Este exemplo cria um robô. Ele usa a ARMHF arquitetura. Ele também anexa uma tag chamada Região.

Comando:

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [CreateRobot](#) na Referência de AWS CLI Comandos.

create-simulation-application-version

O código de exemplo a seguir mostra como usar `create-simulation-application-version`.

AWS CLI

Para criar uma versão do aplicativo de simulação

Este exemplo cria uma versão do aplicativo robótico.

Comando:


```
aws robomaker create-simulation-application-version --  
application arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MySimulationApplication/1551203427605
```

Saída:

```
{  
  "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": {}  
}
```

- Para API obter detalhes, consulte [CreateSimulationApplicationVersion](#) na Referência de AWS CLI Comandos.

create-simulation-application

O código de exemplo a seguir mostra como usar `create-simulation-application`.

AWS CLI

Para criar um aplicativo de simulação

Este exemplo cria um aplicativo de simulação.

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
```

Saída:

```
{  
  "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": {}  
}
```

- Para API obter detalhes, consulte [CreateSimulationApplication](#) na Referência de AWS CLI Comandos.

create-simulation-job

O código de exemplo a seguir mostra como usar create-simulation-job.

AWS CLI

Para criar um trabalho de simulação

Este exemplo cria um trabalho de simulação. Ele usa um aplicativo de robô e um aplicativo de simulação.

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,laun  
--tags Region=North
```

Saída:

```
{  
  "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"
    }
}

```

- Para API obter detalhes, consulte [CreateSimulationJob](#) na Referência de AWS CLI Comandos.

delete-fleet

O código de exemplo a seguir mostra como usar `delete-fleet`.

AWS CLI

Para excluir uma frota

Este exemplo exclui uma frota.

Comando:

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- Para API obter detalhes, consulte [DeleteFleet](#) na Referência de AWS CLI Comandos.

delete-robot-application

O código de exemplo a seguir mostra como usar `delete-robot-application`.

AWS CLI

Para excluir um aplicativo de robô

Este exemplo exclui um aplicativo de robô.

Comando:

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- Para API obter detalhes, consulte [DeleteRobotApplication](#) na Referência de AWS CLI Comandos.

delete-robot

O código de exemplo a seguir mostra como usar `delete-robot`.

AWS CLI

Para excluir um robô

Este exemplo exclui um robô.

Comando:

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778
```

- Para API obter detalhes, consulte [DeleteRobot](#) na Referência de AWS CLI Comandos.

delete-simulation-application

O código de exemplo a seguir mostra como usar `delete-simulation-application`.

AWS CLI

Para excluir um aplicativo de simulação

Este exemplo exclui um aplicativo de simulação.

Comando:

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- Para API obter detalhes, consulte [DeleteSimulationApplication](#) na Referência de AWS CLI Comandos.

deregister-robot

O código de exemplo a seguir mostra como usar `deregister-robot`.

AWS CLI

Para cancelar o registro de um robô de uma frota

Este exemplo cancela o registro de um robô de uma frota.

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
```

Saída:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- Para API obter detalhes, consulte [DeregisterRobot](#) na Referência de AWS CLI Comandos.

describe-deployment-job

O código de exemplo a seguir mostra como usar `describe-deployment-job`.

AWS CLI

Para descrever um trabalho de implantação

O `describe-deployment-job` exemplo a seguir recupera os detalhes sobre o trabalho de implantação especificado.

```
aws robomaker describe-deployment-job \  
  --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-  
xl8qssl6pbcn
```

Saída:

```
{  
  "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": {}  
  }  
}
```

- Para API obter detalhes, consulte [DescribeDeploymentJob](#) na Referência de AWS CLI Comandos.

describe-fleet

O código de exemplo a seguir mostra como usar `describe-fleet`.

AWS CLI

Para descrever uma frota

O `describe-fleet` exemplo a seguir recupera os detalhes da frota especificada.

```
aws robomaker describe-fleet \  
  --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/  
MyFleet/1550771358907
```

Saída:

```
{  
  "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": {}
}
```

- Para API obter detalhes, consulte [DescribeFleet](#) na Referência de AWS CLI Comandos.

describe-robot-application

O código de exemplo a seguir mostra como usar `describe-robot-application`.

AWS CLI

Para descrever uma aplicação robótica

Este exemplo descreve uma aplicação robótica.

Comando:

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

Saída:

```
{
  "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": {}  
}
```

- Para API obter detalhes, consulte [DescribeRobotApplication](#) na Referência de AWS CLI Comandos.

describe-robot

O código de exemplo a seguir mostra como usar `describe-robot`.

AWS CLI

Para descrever um robô

Este exemplo descreve um robô.

Comando:

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/  
MyRobot/1550772324398
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [DescribeRobot](#) na Referência de AWS CLI Comandos.

describe-simulation-application

O código de exemplo a seguir mostra como usar `describe-simulation-application`.

AWS CLI

Para descrever um aplicativo de simulação

Este exemplo descreve um aplicativo de simulação.

Comando:

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

Saída:

```
{
  "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": {}
}
```

- Para API obter detalhes, consulte [DescribeSimulationApplication](#) na Referência de AWS CLI Comandos.

describe-simulation-job

O código de exemplo a seguir mostra como usar `describe-simulation-job`.

AWS CLI

Para descrever um trabalho de simulação

Este exemplo descreve um trabalho de simulação.

Comando:

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

Saída:

```
{
  "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": {}
}
```

- Para API obter detalhes, consulte [DescribeSimulationJob](#) na Referência de AWS CLI Comandos.

list-deployment-jobs

O código de exemplo a seguir mostra como usar `list-deployment-jobs`.

AWS CLI

Para listar trabalhos de implantação

O `list-deployment-jobs` exemplo a seguir recupera uma lista de trabalhos de implantação.

```
aws robomaker list-deployment-jobs
```

Saída:

```
{
  "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
}
]

```

```
}
```

- Para API obter detalhes, consulte [ListDeploymentJobs](#) na Referência de AWS CLI Comandos.

list-fleets

O código de exemplo a seguir mostra como usar `list-fleets`.

AWS CLI

Para listar frotas

Este exemplo lista frotas. No máximo 20 frotas serão devolvidas.

Comando:

```
aws robomaker list-fleets --max-items 20
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListFleets](#) na Referência de AWS CLI Comandos.

list-robot-applications

O código de exemplo a seguir mostra como usar `list-robot-applications`.

AWS CLI

Para listar aplicativos de robôs

Este exemplo lista aplicativos de robôs. Os resultados são limitados a 20 aplicações de robôs.

Comando:

```
aws robomaker list-robot-applications --max-results 20
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListRobotApplications](#) na Referência de AWS CLI Comandos.

list-robots

O código de exemplo a seguir mostra como usar `list-robots`.

AWS CLI

Para listar robôs

Este exemplo lista robôs. No máximo 20 robôs serão devolvidos.

Comando:

```
aws robomaker list-robots --max-results 20
```

Saída:

```
{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListRobots](#) na Referência de AWS CLI Comandos.

list-simulation-applications

O código de exemplo a seguir mostra como usar `list-simulation-applications`.

AWS CLI

Para listar aplicativos de simulação

Este exemplo lista aplicativos de simulação. No máximo 20 aplicativos de simulação serão devolvidos.

Comando:

```
aws robomaker list-simulation-applications --max-results 20
```

Saída:

```

{
  "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_0LIIt6D1h6",
    "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/AWSRoboMakerCloudWatch-1547663411642_0LIIt6D1h6/1547663521470",
    "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
  }
]
}

```

- Para API obter detalhes, consulte [ListSimulationApplications](#) na Referência de AWS CLI Comandos.

list-simulation-jobs

O código de exemplo a seguir mostra como usar `list-simulation-jobs`.

AWS CLI

Para listar trabalhos de simulação

Este exemplo lista trabalhos de simulação.

Comando:

```
aws robomaker list-simulation-jobs
```

Saída:

```
{
  "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"
    ]
}
]
}

```

- Para API obter detalhes, consulte [ListSimulationJobs](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar tags para um recurso

Este exemplo lista as tags de um AWS RoboMaker recurso.

Comando:

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```

Saída:

```
{
  "tags": {
    "Region": "North",
    "Stage": "Initial"
  }
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

register-robot

O código de exemplo a seguir mostra como usar `register-robot`.

AWS CLI

Para registrar um robô

Este exemplo registra um robô em uma frota.

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
```

Saída:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- Para API obter detalhes, consulte [RegisterRobot](#) na Referência de AWS CLI Comandos.

restart-simulation-job

O código de exemplo a seguir mostra como usar `restart-simulation-job`.

AWS CLI

Para reiniciar uma simulação

Este exemplo reinicia uma simulação.

Comando:

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- Para API obter detalhes, consulte [RestartSimulationJob](#) na Referência de AWS CLI Comandos.

sync-deployment-job

O código de exemplo a seguir mostra como usar `sync-deployment-job`.

AWS CLI

Para sincronizar um trabalho de implantação

Este exemplo sincroniza um trabalho de implantação.

Comando:

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/Trek/1539894765711
```

Saída:

```
{
  "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
}
```

- Para API obter detalhes, consulte [SyncDeploymentJob](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para marcar um recurso

Este exemplo marca um recurso. Ele anexa duas tags: Região e Estágio.

Comando:

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para desmarcar um recurso

Este exemplo remove uma tag de um recurso. Ele remove a tag Região.

Comando:

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-robot-application

O código de exemplo a seguir mostra como usar `update-robot-application`.

AWS CLI

Para atualizar um aplicativo de robô

Este exemplo atualiza um aplicativo de robô.

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
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [UpdateRobotApplication](#) na Referência de AWS CLI Comandos.

update-simulation-application

O código de exemplo a seguir mostra como usar `update-simulation-application`.

AWS CLI

Para atualizar um aplicativo de simulação

Este exemplo atualiza um aplicativo de simulação.

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
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [UpdateSimulationApplication](#) na Referência de AWS CLI Comandos.

Exemplos do Route 53 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Route 53.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

change-resource-record-sets

O código de exemplo a seguir mostra como usar `change-resource-record-sets`.

AWS CLI

Para criar, atualizar ou excluir um conjunto de registros de recursos

O `change-resource-record-sets` comando a seguir cria um conjunto de registros de recursos usando a configuração `hosted-zone-id` `Z1R8UBAEXAMPLE` e a JSON formatada no arquivo: `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
```

Para obter mais informações, consulte `POST ChangeResourceRecordSets` na [Amazon Route 53 API Reference](#).

A configuração no JSON arquivo depende do tipo de conjunto de registros de recursos que você deseja criar:

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover Pseudônimo

Sintaxe básica:

```
{
  "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"
          },
          {...}
        ]
      }
    }
  ]
}
```

```

    ]
  }
},
{...}
]
}

```

Sintaxe ponderada:

```

{
  "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"
      }
    },
    {...}
  ]
}

```

Sintaxe do 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"
  }
  },
  {...}
]
}

```

Sintaxe ponderada do alias:

```

{
  "Comment": "optional comment about the changes in this change batch request",
  "Changes": [
    {
      "Action": "CREATE"|"DELETE"|"UPSERT",
      "ResourceRecordSet": {
        "Name": "DNS domain name",
        "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"PTR"|"SRV"|"SPF"|"AAAA",
        "SetIdentifier": "unique description for this resource record set",
        "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"
      }
    },
    {...}
  ]
}

```

Sintaxe de latência:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

Sintaxe do alias de latência:

```
{
  "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"
  }
},
{...}
]
}

```

Sintaxe de 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"
      }
    },
    {...}
  ]
}

```

Sintaxe do alias de 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"
  }
  },
  {...}
]
}

```

- Para API obter detalhes, consulte [ChangeResourceRecordSets](#) na Referência de AWS CLI Comandos.

change-tags-for-resource

O código de exemplo a seguir mostra como usar `change-tags-for-resource`.

AWS CLI

O comando a seguir adiciona uma tag nomeada `owner` a um recurso de verificação de integridade especificado por ID:

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner, Value=myboss
```

O comando a seguir remove uma tag nomeada `owner` de um recurso de zona hospedada especificado por ID:

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-id Z1523434445 --remove-tag-keys owner
```

- Para API obter detalhes, consulte [ChangeTagsForResource](#) na Referência de AWS CLI Comandos.

create-health-check

O código de exemplo a seguir mostra como usar `create-health-check`.

AWS CLI

Para criar uma verificação de saúde

O `create-health-check` comando a seguir cria uma verificação de integridade usando a referência do chamador `2014-04-01-18:47` e a configuração JSON -formatada no arquivo: `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
```

JSONsintaxe:

```
{
  "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
}
```

Para adicionar a verificação de saúde a um conjunto de registros de recursos do Route 53, use o `change-resource-record-sets` comando.

Para obter mais informações, consulte [Amazon Route 53 Health Checks and DNS Failover](#) no [Amazon Route 53 Developer Guide](#).

- Para API obter detalhes, consulte [CreateHealthCheck](#) na Referência de AWS CLI Comandos.

create-hosted-zone

O código de exemplo a seguir mostra como usar `create-hosted-zone`.

AWS CLI

Para criar uma hosted zone

O `create-hosted-zone` comando a seguir adiciona uma zona hospedada chamada `example.com` usando a referência `2014-04-01-18:47` do chamador. O comentário opcional inclui um espaço, portanto, ele deve estar entre aspas:

```
aws route53 create-hosted-zone --name example.com --caller-  
reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

Para obter mais informações, consulte [Como trabalhar com zonas hospedadas](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [CreateHostedZone](#) na Referência de AWS CLI Comandos.

`delete-health-check`

O código de exemplo a seguir mostra como usar `delete-health-check`.

AWS CLI

Para excluir uma verificação de saúde

O `delete-health-check` comando a seguir exclui a verificação de saúde com um `health-check-id` `dee75b48d9-547a-4c3d-88a5-ae4002397608`:

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-  
ae4002397608
```

- Para API obter detalhes, consulte [DeleteHealthCheck](#) na Referência de AWS CLI Comandos.

`delete-hosted-zone`

O código de exemplo a seguir mostra como usar `delete-hosted-zone`.

AWS CLI

Para excluir uma zona hospedada

O `delete-hosted-zone` comando a seguir exclui a zona hospedada com um `id` `deZ36KTIQEXAMPLE`:

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- Para API obter detalhes, consulte [DeleteHostedZone](#) na Referência de AWS CLI Comandos.

get-change

O código de exemplo a seguir mostra como usar get-change.

AWS CLI

Para obter o status de uma alteração nos conjuntos de registros de recursos

O get-change comando a seguir obtém o status e outras informações sobre a change-resource-record-sets solicitação que tem um Id de/change/CWPIK4URU2I5S:

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- Para API obter detalhes, consulte [GetChange](#) na Referência de AWS CLI Comandos.

get-health-check

O código de exemplo a seguir mostra como usar get-health-check.

AWS CLI

Para obter informações sobre uma verificação de saúde

O get-health-check comando a seguir obtém informações sobre a verificação de saúde que tem um health-check-id de02ec8401-9879-4259-91fa-04e66d094674:

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- Para API obter detalhes, consulte [GetHealthCheck](#) na Referência de AWS CLI Comandos.

get-hosted-zone

O código de exemplo a seguir mostra como usar get-hosted-zone.

AWS CLI

Para obter informações sobre uma zona hospedada

O `get-hosted-zone` comando a seguir obtém informações sobre a zona hospedada com um `id` dos `Z1R8UBAEXAMPLE`:

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- Para API obter detalhes, consulte [GetHostedZone](#) na Referência de AWS CLI Comandos.

list-health-checks

O código de exemplo a seguir mostra como usar `list-health-checks`.

AWS CLI

Para listar as verificações de saúde associadas à AWS conta atual

O `list-health-checks` comando a seguir lista informações detalhadas sobre as primeiras 100 verificações de saúde associadas à AWS conta atual. :

```
aws route53 list-health-checks
```

Se você tiver mais de 100 verificações de saúde ou quiser listá-las em grupos menores que 100, inclua o `--max-items` parâmetro. Por exemplo, para listar as verificações de saúde uma por vez, use o seguinte comando:

```
aws route53 list-health-checks --max-items 1
```

Para ver a próxima verificação de saúde, pegue o valor `NextToken` de da resposta ao comando anterior e inclua-o no `--starting-token` parâmetro, por exemplo:

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- Para API obter detalhes, consulte [ListHealthChecks](#) na Referência de AWS CLI Comandos.

list-hosted-zones-by-name

O código de exemplo a seguir mostra como usar `list-hosted-zones-by-name`.

AWS CLI

O comando a seguir lista até 100 zonas hospedadas ordenadas por nome de domínio:

```
aws route53 list-hosted-zones-by-name
```

Saída:

```
{
  "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"
}
```

O comando a seguir lista as zonas hospedadas ordenadas por nome, começando com `www.example.com`:

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [ListHostedZonesByName](#) na Referência de AWS CLI Comandos.

list-hosted-zones

O código de exemplo a seguir mostra como usar `list-hosted-zones`.

AWS CLI

Para listar as zonas hospedadas associadas à AWS conta atual

O `list-hosted-zones` comando a seguir lista informações resumidas sobre as primeiras 100 zonas hospedadas associadas à AWS conta atual. :

```
aws route53 list-hosted-zones
```

Se você tiver mais de 100 zonas hospedadas ou se quiser listá-las em grupos de menos de 100 zonas, inclua o parâmetro `--max-items`. Por exemplo, para listar as zonas hospedadas, use o seguinte comando:

```
aws route53 list-hosted-zones --max-items 1
```

Para visualizar informações sobre a próxima zona hospedada, pegue o valor de `NextToken` da resposta ao comando anterior e inclua-o no parâmetro `--starting-token`, por exemplo:

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- Para API obter detalhes, consulte [ListHostedZones](#) na Referência de AWS CLI Comandos.

list-query-logging-configs

O código de exemplo a seguir mostra como usar `list-query-logging-configs`.

AWS CLI

Para listar as configurações de registro de consultas

O `list-query-logging-configs` exemplo a seguir lista informações sobre as primeiras 100 configurações de registro de consultas em sua AWS conta, para a zona `Z10X3WQEXAMPLE` hospedada.

```
aws route53 list-query-logging-configs \  
  --hosted-zone-id Z10X3WQEXAMPLE
```

Saída:

```
{  
  "QueryLoggingConfigs": [  
    {  
      "Id": "964ff34e-ae03-4f06-80a2-9683cexample",  
      "HostedZoneId": "Z10X3WQEXAMPLE",  
      "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-  
group:/aws/route53/example.com:*"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Registrar DNS consultas](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListQueryLoggingConfigs](#) na Referência de AWS CLI Comandos.

list-resource-record-sets

O código de exemplo a seguir mostra como usar `list-resource-record-sets`.

AWS CLI

Para listar os conjuntos de registros de recursos em uma zona hospedada

O `list-resource-record-sets` comando a seguir lista informações resumidas sobre os primeiros 100 conjuntos de registros de recursos em uma zona hospedada especificada. :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

Se a zona hospedada contiver mais de 100 conjuntos de registros de recursos ou se você quiser listá-los em grupos menores que 100, inclua o `--max-items` parâmetro. Por exemplo, para listar conjuntos de registros de recursos um por vez, use o seguinte comando:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

Para visualizar informações sobre o próximo conjunto de registros de recursos na zona hospedada, use o valor `NextToken` da resposta ao comando anterior e inclua-o no `--starting-token` parâmetro, por exemplo:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

Para visualizar todos os conjuntos de registros de recursos de um nome específico, use o `--query` parâmetro para filtrá-los. Por exemplo:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --query "ResourceRecordSets[?Name == 'example.domain.']"
```

- Para API obter detalhes, consulte [ListResourceRecordSets](#) na Referência de AWS CLI Comandos.

Exemplos de registro de domínio do Route 53 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o registro de domínio AWS Command Line Interface com o Route 53.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

check-domain-availability

O código de exemplo a seguir mostra como usar `check-domain-availability`.

AWS CLI

Para determinar se você pode registrar um nome de domínio com o Route 53

O `check-domain-availability` comando a seguir retorna informações sobre se o nome de domínio `example.com` está disponível para ser registrado usando o Route 53.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains check-domain-availability \
  --region us-east-1 \
  --domain-name example.com
```

Saída:

```
{
  "Availability": "UNAVAILABLE"
}
```

O Route 53 oferece suporte a um grande número de domínios de primeiro nível (TLDs), como `.com` e `.jp`, mas não oferecemos suporte a todos os disponíveis. TLDs Se você verificar a disponibilidade de um domínio e o Route 53 não suportar o TLD, `check-domain-availability` retornará a seguinte mensagem.

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability
operation: <top-level domain> tld is not supported.
```

Para obter uma lista dos TLDs que você pode usar ao registrar um domínio no Route 53, consulte [Domains That You Can Register with Amazon Route 53 no Amazon Route 53 Developer Guide](#). Para obter mais informações sobre o registro de domínios no Amazon Route 53, consulte [Registro de um novo domínio](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [CheckDomainAvailability](#) na Referência de AWS CLI Comandos.

check-domain-transferability

O código de exemplo a seguir mostra como usar `check-domain-transferability`.

AWS CLI

Para determinar se um domínio pode ser transferido para o Route 53

O `check-domain-transferability` comando a seguir retorna informações sobre se você pode transferir o nome de domínio `example.com` para o Route 53.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains check-domain-transferability \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "Transferability": {  
    "Transferable": "UNTRANSFERABLE"  
  }  
}
```

Para obter mais informações, consulte [Transferência do registro de um domínio para o Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [CheckDomainTransferability](#) na Referência de AWS CLI Comandos.

delete-tags-for-domain

O código de exemplo a seguir mostra como usar `delete-tags-for-domain`.

AWS CLI

Para excluir tags de um domínio

O `delete-tags-for-domain` comando a seguir exclui três tags do domínio especificado. Observe que você especifica somente a chave da tag, não o valor da tag.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains delete-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --tags-to-delete accounting-key hr-key engineering-key
```

Este comando não produz saída.

Para confirmar que as tags foram excluídas, você pode executar [list-tags-for-domain](#). Para obter mais informações, consulte Como [marcar os recursos do Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [DeleteTagsForDomain](#) na Referência de AWS CLI Comandos.

disable-domain-auto-renew

O código de exemplo a seguir mostra como usar `disable-domain-auto-renew`.

AWS CLI

Para desativar a renovação automática de um domínio

O `disable-domain-auto-renew` comando a seguir configura o Route 53 para não renovar automaticamente o domínio `example.com` antes que o registro do domínio expire.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains disable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

Este comando não produz saída.

Para confirmar que a configuração foi alterada, você pode executar [get-domain-detail](#). Se a renovação automática estiver desativada, o valor de `AutoRenew` é `False`. Para obter mais informações sobre renovação automática, consulte [Renovação do registro de um domínio < https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [DisableDomainAutoRenew](#) na Referência de AWS CLI Comandos.

disable-domain-transfer-lock

O código de exemplo a seguir mostra como usar `disable-domain-transfer-lock`.

AWS CLI

Para desativar o bloqueio de transferência em um domínio

O `disable-domain-transfer-lock` comando a seguir remove o bloqueio de transferência no domínio `example.com` para que o domínio possa ser transferido para outro registrador. Esse comando altera o `clientTransferProhibited` status.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains disable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Para confirmar que o bloqueio de transferência foi alterado, você pode executar [get-domain-detail](#). Quando o bloqueio de transferência está desativado, o valor de `StatusList` não é `includoclientTransferProhibited`.

Para obter mais informações sobre o processo de transferência, consulte [Transferência de um domínio do Amazon Route 53 para outro registrador](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [DisableDomainTransferLock](#) na Referência de AWS CLI Comandos.

enable-domain-auto-renew

O código de exemplo a seguir mostra como usar `enable-domain-auto-renew`.

AWS CLI

Para ativar a renovação automática de um domínio

O `enable-domain-auto-renew` comando a seguir configura o Route 53 para renovar automaticamente o domínio `example.com` antes que o registro do domínio expire.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains enable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

Este comando não produz saída. Para confirmar que a configuração foi alterada, você pode executar [get-domain-detail](#). Se a renovação automática estiver ativada, o valor de `AutoRenew` é `True`.

Para obter mais informações sobre renovação automática, consulte [Renovação do registro de um domínio < https://docs.aws.amazon.com/Route53/latest/ DeveloperGuide /domain-renew.html](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [EnableDomainAutoRenew](#) na Referência de AWS CLI Comandos.

enable-domain-transfer-lock

O código de exemplo a seguir mostra como usar `enable-domain-transfer-lock`.

AWS CLI

Para ativar o bloqueio de transferência em um domínio

O `enable-domain-transfer-lock` comando a seguir bloqueia o domínio especificado para que ele não possa ser transferido para outro registrador. Esse comando altera o `clientTransferProhibited` status.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida com `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains enable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Para confirmar que o bloqueio de transferência foi alterado, você pode executar [get-domain-detail](#). Quando o bloqueio de transferência está ativado, o valor de `StatusList` inclui `clientTransferProhibited`.

Para obter mais informações sobre o processo de transferência, consulte [Transferência de um domínio do Amazon Route 53 para outro registrador](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [EnableDomainTransferLock](#) na Referência de AWS CLI Comandos.

get-contact-reachability-status

O código de exemplo a seguir mostra como usar `get-contact-reachability-status`.

AWS CLI

Para determinar se o contato do registrante respondeu a um e-mail de confirmação

O `get-contact-reachability-status` comando a seguir retorna informações sobre se o contato do solicitante do registro do domínio especificado respondeu a um e-mail de confirmação.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains get-contact-reachability-status \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "domainName": "example.com",  
  "status": "DONE"  
}
```

Para obter mais informações, consulte [Reenvio de e-mails de autorização e confirmação](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [GetContactReachabilityStatus](#) na Referência de AWS CLI Comandos.

get-domain-detail

O código de exemplo a seguir mostra como usar `get-domain-detail`.

AWS CLI

Para obter informações detalhadas sobre um domínio especificado

O `get-domain-detail` comando a seguir exibe informações detalhadas sobre o domínio especificado.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.


```
aws route53domains get-domain-detail \  
--region us-east-1 \  
--domain-name example.com
```

Saída:

```
{  
  "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"
  ]
}
```

- Para API obter detalhes, consulte [GetDomainDetail](#) na Referência de AWS CLI Comandos.

get-domain-suggestions

O código de exemplo a seguir mostra como usar `get-domain-suggestions`.

AWS CLI

Para obter uma lista de nomes de domínio sugeridos

O `get-domain-suggestions` comando a seguir exibe uma lista de nomes de domínio sugeridos com base no nome do domínio `example.com`. A resposta inclui somente nomes de domínio que estão disponíveis. Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains get-domain-suggestions \
  --region us-east-1 \
  --domain-name example.com \
  --suggestion-count 10 \
  --only-available
```

Saída:

```
{
  "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": "exemplenews.net",
      "Availability": "AVAILABLE"
    },
    {
      "DomainName": "officeexample.com",
      "Availability": "AVAILABLE"
    },
    {
      "DomainName": "exampleworld.com",
      "Availability": "AVAILABLE"
    },
    {
      "DomainName": "exampleart.com",
      "Availability": "AVAILABLE"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetDomainSuggestions](#) na Referência de AWS CLI Comandos.

get-operation-detail

O código de exemplo a seguir mostra como usar `get-operation-detail`.

AWS CLI

Para obter o status atual de uma operação

Algumas operações de registro de domínio operam de forma assíncrona e retornam uma resposta antes de serem concluídas. Essas operações retornam um ID de operação que você pode usar para obter o status atual. O `get-operation-detail` comando a seguir retorna o status da operação especificada.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains get-operation-detail \
  --region us-east-1 \
```

```
--operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

Saída:

```
{
  "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",
  "Status": "SUCCESSFUL",
  "DomainName": "example.com",
  "Type": "DOMAIN_LOCK",
  "SubmittedDate": 1573749367.864
}
```

- Para API obter detalhes, consulte [GetOperationDetail](#) na Referência de AWS CLI Comandos.

list-domains

O código de exemplo a seguir mostra como usar `list-domains`.

AWS CLI

Para listar os domínios que estão registrados com a conta atual AWS

O `list-domains` comando a seguir lista informações resumidas sobre os domínios registrados na AWS conta atual.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains list-domains
  --region us-east-1
```

Saída:

```
{
  "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
}
]
```

- Para API obter detalhes, consulte [ListDomains](#) na Referência de AWS CLI Comandos.

list-operations

O código de exemplo a seguir mostra como usar `list-operations`.

AWS CLI

Para listar o status das operações que retornam um ID de operação

Algumas operações de registro de domínio são executadas de forma assíncrona e retornam uma resposta antes de serem concluídas. Essas operações retornam um ID de operação que você pode usar para obter o status atual. O `list-operations` comando a seguir lista informações resumidas, incluindo o status, sobre as operações atuais de registro de domínio.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains list-operations
--region us-east-1
```

Saída:

```
{
  "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
  }
]
}

```

A saída inclui todas as operações que retornam um ID de operação e que você executou em todos os domínios que você já registrou usando a AWS conta atual. Se quiser obter somente as operações enviadas após uma data especificada, você pode incluir o `submitted-since` parâmetro e especificar uma data no formato Unix e no Tempo Universal Coordenado (UTC). O comando a seguir obtém o status de todas as operações enviadas após as 12h do dia UTC 1º de janeiro de 2020.

```

aws route53domains list-operations \
  --submitted-since 1577836800

```

- Para API obter detalhes, consulte [ListOperations](#) na Referência de AWS CLI Comandos.

list-tags-for-domain

O código de exemplo a seguir mostra como usar `list-tags-for-domain`.

AWS CLI

Para listar tags para um domínio

O `list-tags-for-domain` comando a seguir lista as tags atualmente associadas ao domínio especificado.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains list-tags-for-domain \
  --region us-east-1 \
  --domain-name example.com
```

Saída:

```
{
  "TagList": [
    {
      "Key": "key1",
      "Value": "value1"
    },
    {
      "Key": "key2",
      "Value": "value2"
    }
  ]
}
```

Para obter mais informações, consulte Como [marcar os recursos do Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [ListTagsForDomain](#) na Referência de AWS CLI Comandos.

register-domain

O código de exemplo a seguir mostra como usar `register-domain`.

AWS CLI

Para registrar um domínio

O `register-domain` comando a seguir registra um domínio, recuperando todos os valores dos parâmetros de um arquivo JSON formatado.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains register-domain \  
  --region us-east-1 \  
  --cli-input-json file://register-domain.json
```

Conteúdo de `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
}
```

Saída:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation-detail`. Para obter mais informações, consulte [get-operation-detail](#).

Para obter mais informações, consulte [Registrar um novo domínio](#) no Guia do desenvolvedor do Amazon Route 53.

Para obter informações sobre quais domínios de primeiro nível (TLDs) exigem valores `ExtraParams` e quais são os valores válidos, consulte [ExtraParama](#) Referência do Amazon Route 53 API.

- Para API obter detalhes, consulte [RegisterDomain](#) na Referência de AWS CLI Comandos.

renew-domain

O código de exemplo a seguir mostra como usar `renew-domain`.

AWS CLI

Para renovar um domínio

O `renew-domain` comando a seguir renova o domínio especificado por cinco anos. Para obter o valor `current-expiry-year`, use o `get-domain-detail` comando e converta o valor de `ExpirationDate` do formato Unix.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains renew-domain \
  --region us-east-1 \
  --domain-name example.com \
  --duration-in-years 5 \
  --current-expiry-year 2020
```

Saída:

```
{
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation-detail`. Para obter mais informações, consulte [get-operation-detail](#).

O registro de cada domínio de primeiro nível (TLD), como `.com` ou `.org`, controla o número máximo de anos pelos quais você pode renovar um domínio. Para obter o período máximo de renovação do seu domínio, consulte a seção “Período de registro e renovação” dos seus [domínios que você pode registrar TLD no Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

Para obter mais informações, consulte [Renovação do registro de um domínio](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [RenewDomain](#) na Referência de AWS CLI Comandos.

resend-contact-reachability-email

O código de exemplo a seguir mostra como usar `resend-contact-reachability-email`.

AWS CLI

Para reenviar o e-mail de confirmação para o endereço de e-mail atual do registrante, entre em contato

O `resend-contact-reachability-email` comando a seguir reenvia o e-mail de confirmação para o endereço de e-mail atual do contato do registrante do domínio `example.com`.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains resend-contact-reachability-email \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "domainName": "example.com",  
  "emailAddress": "moliveira@example.com",  
  "isAlreadyVerified": true  
}
```

Se o valor de `isAlreadyVerified` for `true`, como neste exemplo, o contato do registrante já confirmou que o endereço de e-mail especificado está acessível.

Para obter mais informações, consulte [Reenvio de e-mails de autorização e confirmação](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [ResendContactReachabilityEmail](#) na Referência de AWS CLI Comandos.

retrieve-domain-auth-code

O código de exemplo a seguir mostra como usar `retrieve-domain-auth-code`.

AWS CLI

Para obter o código de autorização de um domínio para que você possa transferir o domínio para outro registrador

O `retrieve-domain-auth-code` comando a seguir obtém o código de autorização atual para o domínio `example.com`. Você atribui esse valor a outro registrador de domínio quando deseja transferir o domínio para esse registrador.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains retrieve-domain-auth-code \  
  --region us-east-1 \  
  --domain-name example.com
```

Saída:

```
{  
  "AuthCode": ")o!v3dJeXampLe"  
}
```

Para obter mais informações, consulte [Transferência de um domínio do Amazon Route 53 para outro registrador](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [RetrieveDomainAuthCode](#) na Referência de AWS CLI Comandos.

transfer-domain

O código de exemplo a seguir mostra como usar `transfer-domain`.

AWS CLI

Para transferir um domínio para o Amazon Route 53

O `transfer-domain` comando a seguir transfere um domínio para o Route 53, com os parâmetros fornecidos pelo arquivo JSON `-formatted`. `C:\temp\transfer-domain.json`

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains transfer-domain \  
  --region us-east-1 \  
  --cli-input-json file://C:\temp\transfer-domain.json
```

Conteúdo de `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
}
```

Saída:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation-detail`. Para obter mais informações, consulte [get-operation-detail](#).

Para obter mais informações, consulte [Transferência do registro de um domínio para o Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [TransferDomain](#) na Referência de AWS CLI Comandos.

update-domain-contact-privacy

O código de exemplo a seguir mostra como usar `update-domain-contact-privacy`.

AWS CLI

Para atualizar as configurações de privacidade dos contatos de um domínio

O `update-domain-contact-privacy` comando a seguir desativa a proteção de privacidade do contato administrativo do domínio `example.com`. Esse comando é executado somente na `us-east-1` região.

Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains update-domain-contact-privacy \
```

```
--region us-east-1 \  
--domain-name example.com \  
--no-admin-privacy
```

Saída:

```
{  
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"  
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar `target-operation-detail`. Para obter mais informações, consulte [get-operation-detail](#).

Para obter mais informações, consulte [Ativar ou desativar a proteção de privacidade para informações de contato de um domínio](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [UpdateDomainContactPrivacy](#) na Referência de AWS CLI Comandos.

update-domain-contact

O código de exemplo a seguir mostra como usar `update-domain-contact`.

AWS CLI

Para atualizar as informações de contato de um domínio

O `update-domain-contact` comando a seguir atualiza as informações de contato de um domínio, obtendo os parâmetros do arquivo JSON -formatado. `C:\temp\update-domain-contact.json`

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains update-domain-contact \  
  --region us-east-1 \  
  --cli-input-json file://C:\temp\update-domain-contact.json
```

Conteúdo de `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"
  }
}
```

Saída:

```
{
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar [get-domain-detail](#). Para obter mais informações, consulte [Atualização das informações de contato de um domínio](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [UpdateDomainContact](#) na Referência de AWS CLI Comandos.

update-domain-nameservers

O código de exemplo a seguir mostra como usar `update-domain-nameservers`.

AWS CLI

Para atualizar os servidores de nomes de um domínio

O `update-domain-nameservers` comando a seguir atualiza os servidores de nomes de um domínio.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
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=ns-4.awsdns-04.com
```

Saída:

```
{
  "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

Para confirmar que a operação foi bem-sucedida, você pode executar [get-domain-detail](#).

Para obter mais informações, consulte [Adicionar ou alterar servidores de nomes e registros Glue para um domínio](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [UpdateDomainNameservers](#) na Referência de AWS CLI Comandos.

update-tags-for-domain

O código de exemplo a seguir mostra como usar `update-tags-for-domain`.

AWS CLI

Para adicionar ou atualizar tags para um domínio

O `update-tags-for-domain` comando a seguir adiciona ou atualiza duas chaves e os valores correspondentes para o domínio `example.com`. Para atualizar o valor de uma chave, basta incluir a chave e o novo valor. Você pode adicionar ou atualizar tags em apenas um domínio por vez.

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
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"
```

Este comando não produz saída. Para confirmar se as tags foram adicionadas ou atualizadas, você pode executar [list-tags-for-domain](#).

Para obter mais informações, consulte Como [marcar os recursos do Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [UpdateTagsForDomain](#) na Referência de AWS CLI Comandos.

view-billing

O código de exemplo a seguir mostra como usar `view-billing`.

AWS CLI

Para obter informações de cobrança das cobranças de registro de domínio da conta corrente AWS

O `view-billing` comando a seguir retorna todos os registros de cobrança relacionados ao domínio da conta corrente no período de 1º de janeiro de 2018 (1514764800 no horário Unix) e meia-noite de 31 de dezembro de 2019 (1577836800 no horário Unix).

Esse comando é executado somente na `us-east-1` região. Se sua região padrão estiver definida como `us-east-1`, você poderá omitir o `region` parâmetro.

```
aws route53domains view-billing \  
  --region us-east-1 \  
  --start-time 1514764800 \  
  --end-time 1577836800
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [ViewBilling](#) na Amazon Route 53 API Reference.

- Para API obter detalhes, consulte [ViewBilling](#) na Referência de AWS CLI Comandos.

Exemplos de resolvedores do Route 53 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o Resolvedor do AWS Command Line Interface Route 53.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-firewall-rule-group

O código de exemplo a seguir mostra como usar `associate-firewall-rule-group`.

AWS CLI

Para associar um grupo de regras de firewall a um VPC

O `associate-firewall-rule-group` exemplo a seguir associa um grupo de regras de DNS firewall a uma AmazonVPC.

```
aws route53resolver associate-firewall-rule-group \  
  --name test-association \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --vpc-id vpc-31e92222 \  
  --priority 101
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciando associações entre seus grupos de regras VPC e do Route 53 Resolver DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [AssociateFirewallRuleGroup](#) na Referência de AWS CLI Comandos.

associate-resolver-endpoint-ip-address

O código de exemplo a seguir mostra como usar `associate-resolver-endpoint-ip-address`.

AWS CLI

Para associar outro endereço IP a um endpoint do Resolver

O `associate-resolver-endpoint-ip-address` exemplo a seguir associa outro endereço IP a um endpoint Resolver de entrada. Se você especificar somente uma ID de sub-rede e omitir o endereço IP do `--ip-address` parâmetro, o Resolver escolherá um endereço IP para você dentre os endereços IP disponíveis na sub-rede especificada.

```

aws route53resolver associate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-497098ad5example \
  --ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"

```

Saída:

```
{
```

```

    "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"
    }
  }
}

```

Para obter mais informações, consulte [Valores que você especifica ao criar ou editar endpoints de entrada](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [AssociateResolverEndpointIpAddress](#) na Referência de AWS CLI Comandos.

associate-resolver-rule

O código de exemplo a seguir mostra como usar `associate-resolver-rule`.

AWS CLI

Para associar uma regra do Resolver a um VPC

O `associate-resolver-rule` exemplo a seguir associa uma regra Resolver a uma AmazonVPC. Depois de executar o comando, o Resolver começa a encaminhar DNS consultas para sua rede com base nas configurações da regra, como o nome de domínio das consultas que são encaminhadas.

```

aws route53resolver associate-resolver-rule \
  --name my-resolver-rule-association \
  --resolver-rule-id rslvr-rr-42b60677c0example \
  --vpc-id vpc-304bexam

```

Saída:

```
{
  "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."
  }
}
```

Para obter mais informações, consulte [Encaminhando DNS consultas externas para sua rede](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [AssociateResolverRule](#) na Referência de AWS CLI Comandos.

create-firewall-domain-list

O código de exemplo a seguir mostra como usar `create-firewall-domain-list`.

AWS CLI

Para criar uma lista de domínios do Route 53 Resolver DNS Firewall

O `create-firewall-domain-list` exemplo a seguir cria uma lista de domínios do Route 53 Resolver DNS Firewall, chamada `test`, em sua AWS conta.

```
aws route53resolver create-firewall-domain-list \
  --creator-request-id my-request-id \
  --name test
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciando suas próprias listas de domínios](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [CreateFirewallDomainList](#) na Referência de AWS CLI Comandos.

create-firewall-rule-group

O código de exemplo a seguir mostra como usar `create-firewall-rule-group`.

AWS CLI

Para criar um grupo de regras de firewall

O `create-firewall-rule-group` exemplo a seguir cria um grupo de regras de DNS firewall.

```

aws route53resolver create-firewall-rule-group \
  --creator-request-id my-request-id \
  --name test

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [CreateFirewallRuleGroup](#) na Referência de AWS CLI Comandos.

create-firewall-rule

O código de exemplo a seguir mostra como usar `create-firewall-rule`.

AWS CLI

Para criar uma regra de firewall

O `create-firewall-rule` exemplo a seguir cria uma regra de firewall em uma regra de DNS firewall para domínios listados em uma lista de domínios de 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

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [CreateFirewallRule](#) na Referência de AWS CLI Comandos.

create-resolver-endpoint

O código de exemplo a seguir mostra como usar `create-resolver-endpoint`.

AWS CLI

Para criar um endpoint de entrada do Resolver

O `create-resolver-endpoint` exemplo a seguir cria um endpoint Resolver de entrada. Você pode usar o mesmo comando para criar pontos de extremidade de entrada e saída.

```
aws route53resolver create-resolver-endpoint --name my-inbound-endpoint -- creator-request-id 2020-01-01-18:47 -- security-group-ids "sg-f62bexam" --direction --ip-addresses =subnet-ba47exam, Ip=192.0.2.255 =subnet-12d8exam, Ip=192.0.2.254 INBOUND SubnetId SubnetId
```

Saída:

```

{
  "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"  
  }  
}
```

Para criar um endpoint de saída do Resolver

O `create-resolver-endpoint` exemplo a seguir cria um ponto final do resolvidor de saída usando os valores no documento JSON formatado. `create-outbound-resolver-endpoint.json`

```
aws route53resolver create-resolver-endpoint \  
  --cli-input-json file://c:\temp\create-outbound-resolver-endpoint.json
```

Conteúdo de `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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Resolvendo DNS consultas entre VPCs e sua rede no Guia](#) do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [CreateResolverEndpoint](#) na Referência de AWS CLI Comandos.

create-resolver-rule

O código de exemplo a seguir mostra como usar `create-resolver-rule`.

AWS CLI

Para criar uma regra de resolução

O `create-resolver-rule` exemplo a seguir cria uma regra de encaminhamento do Resolver. A regra usa o endpoint de saída `rslvr-out-d5e5920e37example` para encaminhar DNS consultas para os endereços IP `10.24.8.75` e `10.24.8.156`. `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"
```

Saída:

```
{  
  "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."
}
}

```

- Para API obter detalhes, consulte [CreateResolverRule](#) na Referência de AWS CLI Comandos.

delete-firewall-domain-list

O código de exemplo a seguir mostra como usar `delete-firewall-domain-list`.

AWS CLI

Para excluir uma lista de domínios do Route 53 Resolver DNS Firewall

O `delete-firewall-domain-list` exemplo a seguir exclui uma lista de domínios do Route 53 Resolver DNS Firewall, chamada `test`, em sua AWS conta.

```

aws route53resolver delete-firewall-domain-list \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample

```

Saída:

```

{
  "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"
  }
}
```

Para obter mais informações, consulte [Gerenciando suas próprias listas de domínios](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [DeleteFirewallDomainList](#) na Referência de AWS CLI Comandos.

delete-firewall-rule-group

O código de exemplo a seguir mostra como usar `delete-firewall-rule-group`.

AWS CLI

Para excluir um grupo de regras de firewall

O `delete-firewall-rule-group` exemplo a seguir exclui um grupo de regras de firewall.

```
aws route53resolver delete-firewall-rule-group \
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Saída:

```
{
  "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"
  }
}
```

```
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [DeleteFirewallRuleGroup](#) na Referência de AWS CLI Comandos.

delete-firewall-rule

O código de exemplo a seguir mostra como usar `delete-firewall-rule`.

AWS CLI

Para excluir uma regra de firewall

O `delete-firewall-rule` exemplo a seguir exclui uma regra de firewall especificada.

```
aws route53resolver delete-firewall-rule \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [DeleteFirewallRule](#) na Referência de AWS CLI Comandos.

delete-resolver-endpoint

O código de exemplo a seguir mostra como usar `delete-resolver-endpoint`.

AWS CLI

Para excluir um endpoint do Resolver

O `delete-resolver-endpoint` exemplo a seguir exclui o endpoint especificado.

Importante Se você excluir um endpoint de entrada, DNS as consultas da sua rede não serão mais encaminhadas para o Resolver no VPC que você especificou no endpoint. Se você excluir um endpoint de saída, o Resolver interromperá o encaminhamento de DNS consultas de você VPC para sua rede para regras que especificam o endpoint de saída excluído.

```
aws route53resolver delete-resolver-endpoint \  
  --resolver-endpoint-id rslvr-in-497098ad59example
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [DeleteResolverEndpoint](#) na Referência de AWS CLI Comandos.

delete-resolver-rule

O código de exemplo a seguir mostra como usar `delete-resolver-rule`.

AWS CLI

Para excluir uma regra do Resolver

O `delete-resolver-rule` exemplo a seguir exclui a regra especificada.

Nota Se uma regra estiver associada a alguma VPCs, você deve primeiro desassociar a regra da VPCs antes de poder excluí-la.

```
aws route53resolver delete-resolver-rule \  
  --resolver-rule-id rslvr-rr-5b3809426bexample
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [DeleteResolverRule](#) na Referência de AWS CLI Comandos.

disassociate-firewall-rule-group

O código de exemplo a seguir mostra como usar `disassociate-firewall-rule-group`.

AWS CLI

Para desassociar um grupo de regras de firewall de um VPC

O `disassociate-firewall-rule-group` exemplo a seguir desassocia um grupo de regras de DNS Firewall de uma Amazon. VPC

```
aws route53resolver disassociate-firewall-rule-group \  
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Gerenciando associações entre seus grupos de regras VPC e do Route 53 Resolver DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [DisassociateFirewallRuleGroup](#) na Referência de AWS CLI Comandos.

disassociate-resolver-endpoint-ip-address

O código de exemplo a seguir mostra como usar `disassociate-resolver-endpoint-ip-address`.

AWS CLI

Para desassociar um endereço IP de um endpoint do Resolver

O `disassociate-resolver-endpoint-ip-address` exemplo a seguir remove um endereço IP de um ponto final de entrada ou saída do Resolver especificado.

Observação Um endpoint deve ter pelo menos dois endereços IP. Se um endpoint atualmente tiver apenas dois endereços IP e você quiser substituir um endereço por outro, primeiro use [associate-resolver-endpoint-ip-address](#) para associar o novo endereço IP. Em seguida, você pode desassociar um dos endereços IP originais do endpoint.

```
aws route53resolver disassociate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example \
  --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [DisassociateResolverEndpointIpAddress](#) na Referência de AWS CLI Comandos.

disassociate-resolver-rule

O código de exemplo a seguir mostra como usar `disassociate-resolver-rule`.

AWS CLI

Para desassociar uma regra Resolver de uma Amazon VPC

O `disassociate-resolver-rule` exemplo a seguir remove a associação entre a regra Resolver especificada e a especificada VPC. Você pode dissociar uma regra de a VPC nas seguintes circunstâncias:

Para DNS consultas originadas nissoVPC, você deseja que o Resolver pare de encaminhar consultas para sua rede para o nome de domínio especificado na regra. Você deseja excluir a regra de encaminhamento. Se uma regra estiver atualmente associada a uma ou maisVPCs, você deverá desassociar a regra de todas VPCs antes de poder excluí-la.

```
aws route53resolver disassociate-resolver-rule \  
  --resolver-rule-id rslvr-rr-4955cb98ceexample \  
  --vpc-id vpc-304bexam
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [DisassociateResolverRule](#) na Referência de AWS CLI Comandos.

get-firewall-config

O código de exemplo a seguir mostra como usar `get-firewall-config`.

AWS CLI

Para obter uma configuração de firewall para um VPC

O `get-firewall-config` exemplo a seguir recupera o comportamento do DNS Firewall para o especificado VPC.

```
aws route53resolver get-firewall-config \  
  --resource-id vpc-31e92222
```

Saída:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e92222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

Para obter mais informações, consulte [VPCConfiguração do DNS firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [GetFirewallConfig](#) na Referência de AWS CLI Comandos.

get-firewall-domain-list

O código de exemplo a seguir mostra como usar `get-firewall-domain-list`.

AWS CLI

Para obter uma lista de domínios do Route 53 Resolver DNS Firewall

O `get-firewall-domain-list` exemplo a seguir recupera a lista de domínios com a ID especificada.

```
aws route53resolver get-firewall-domain-list \  
  --resource-id vpc-31e92222
```

```
--firewall-domain-list-id rslvr-fdl-42b60677cexample
```

Saída:

```
{
  "FirewallDomainList": {
    "Id": "rslvr-fdl-9e956e9ffexample",
    "Arn": "arn:aws:route53resolver:us-west-2:123457689012:firewall-domain-list/
rslvr-fdl-42b60677cexample",
    "Name": "test",
    "DomainCount": 0,
    "Status": "COMPLETE",
    "StatusMessage": "Created Firewall Domain List",
    "CreatorRequestId": "my-request-id",
    "CreationTime": "2021-05-25T15:55:51.115365Z",
    "ModificationTime": "2021-05-25T15:55:51.115365Z"
  }
}
```

Para obter mais informações, consulte [Gerenciando suas próprias listas de domínios](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [GetFirewallDomainList](#) na Referência de AWS CLI Comandos.

get-firewall-rule-group-association

O código de exemplo a seguir mostra como usar `get-firewall-rule-group-association`.

AWS CLI

Para obter uma associação de grupo de regras de firewall

O `get-firewall-rule-group-association` exemplo a seguir recupera uma associação de grupo de regras de firewall.

```
aws route53resolver get-firewall-rule-group-association \
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

Saída:

```
{
```

```

    "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"
    }
  }
}

```

Para obter mais informações, consulte [Gerenciando associações entre seus grupos de regras VPC e do Route 53 Resolver DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [GetFirewallRuleGroupAssociation](#) na Referência de AWS CLI Comandos.

get-firewall-rule-group-policy

O código de exemplo a seguir mostra como usar `get-firewall-rule-group-policy`.

AWS CLI

Para obter uma AWS IAM política

O `get-firewall-rule-group-policy` exemplo a seguir obtém a política AWS Identity and Access Management (AWS IAM) para compartilhar o grupo de regras especificado.

```

aws route53resolver get-firewall-rule-group-policy \
  --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/
rslvr-frg-47f93271fexample

```

Saída:

```
{
```



```
"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\" } ] }"
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [GetFirewallRuleGroupPolicy](#) na Referência de AWS CLI Comandos.

get-firewall-rule-group

O código de exemplo a seguir mostra como usar `get-firewall-rule-group`.

AWS CLI

Para obter um grupo de regras de firewall

O `get-firewall-rule-group` exemplo a seguir recupera informações sobre um grupo de regras de DNS firewall com a ID fornecida por você.

```
aws route53resolver get-firewall-rule-group \
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [GetFirewallRuleGroup](#) na Referência de AWS CLI Comandos.

get-resolver-endpoint

O código de exemplo a seguir mostra como usar `get-resolver-endpoint`.

AWS CLI

Para obter informações sobre um endpoint do Resolver

O `get-resolver-endpoint` exemplo a seguir exibe detalhes do endpoint de saída especificado. Você pode usar `get-resolver-endpoint` para endpoints de entrada e saída especificando a ID do endpoint aplicável.

```

aws route53resolver get-resolver-endpoint \
  --resolver-endpoint-id rslvr-out-d5e5920e37example

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Valores que você especifica ao criar ou editar endpoints de entrada](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [GetResolverEndpoint](#) na Referência de AWS CLI Comandos.

get-resolver-rule-association

O código de exemplo a seguir mostra como usar `get-resolver-rule-association`.

AWS CLI

Para obter informações sobre a associação entre uma regra Resolver e uma VPC

O `get-resolver-rule-association` exemplo a seguir exibe detalhes sobre a associação entre uma regra de Resolver especificada e uma VPC. Você associa uma regra de resolução e um VPC uso [associate-resolver-rule](#).

```

aws route53resolver get-resolver-rule-association \
  --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample

```

Saída:

```

{
  "ResolverRuleAssociation": {
    "Id": "rslvr-rrassoc-d61cbb2c8bexample",
    "ResolverRuleId": "rslvr-rr-42b60677c0example",
    "Name": "my-resolver-rule-association",
    "VPCId": "vpc-304bexam",
    "Status": "COMPLETE",
    "StatusMessage": ""
  }
}

```

- Para API obter detalhes, consulte [GetResolverRuleAssociation](#) na Referência de AWS CLI Comandos.

get-resolver-rule

O código de exemplo a seguir mostra como usar `get-resolver-rule`.

AWS CLI

Para obter informações sobre uma regra do Resolver

O `get-resolver-rule` exemplo a seguir exibe detalhes sobre a regra de resolução especificada, como o nome de domínio para o qual a regra encaminha DNS consultas e a ID do endpoint do resolvidor de saída ao qual a regra está associada.

```
aws route53resolver get-resolver-rule \  
  --resolver-rule-id rslvr-rr-42b60677c0example
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Valores que você especifica ao criar ou editar regras](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [GetResolverRule](#) na Referência de AWS CLI Comandos.

import-firewall-domains

O código de exemplo a seguir mostra como usar `import-firewall-domains`.

AWS CLI

Para importar domínios para uma lista de domínios

O `import-firewall-domains` exemplo a seguir importa um conjunto de domínios de um arquivo para uma lista de domínios DNS do Firewall especificada por você.

```
aws route53resolver import-firewall-domains \  
  --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample \  
  --operation REPLACE \  
  --domain-file-url s3://PATH/TO/YOUR/FILE
```

Saída:

```
{  
  "Id": "rslvr-fdl-d61cbb2cbexample",  
  "Name": "test",  
  "Status": "IMPORTING",  
  "StatusMessage": "Importing domains from provided file."  
}
```

Para obter mais informações, consulte [Gerenciando suas próprias listas de domínios](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ImportFirewallDomains](#) na Referência de AWS CLI Comandos.

list-firewall-configs

O código de exemplo a seguir mostra como usar `list-firewall-configs`.

AWS CLI

Para listar as configurações de firewall

O `list-firewall-configs` exemplo a seguir lista suas configurações de DNS firewall.

```
aws route53resolver list-firewall-configs
```

Saída:

```
{
  "FirewallConfigs": [
    {
      "Id": "rslvr-fc-86016850cexample",
      "ResourceId": "vpc-31e92222",
      "OwnerId": "123456789012",
      "FirewallFailOpen": "DISABLED"
    }
  ]
}
```

Para obter mais informações, consulte [VPCConfiguração do DNS firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListFirewallConfigs](#) na Referência de AWS CLI Comandos.

list-firewall-domain-lists

O código de exemplo a seguir mostra como usar `list-firewall-domain-lists`.

AWS CLI

Para listar todas as listas de domínios do Route 53 Resolver DNS Firewall

O `list-firewall-domain-lists` exemplo a seguir lista todas as listas de domínios.

```
aws route53resolver list-firewall-domain-lists
```

Saída:

```
{
  "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"
  }
]
}

```

Para obter mais informações, consulte as [listas de domínios do Route 53 Resolver DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListFirewallDomainLists](#) na Referência de AWS CLI Comandos.

list-firewall-domains

O código de exemplo a seguir mostra como usar `list-firewall-domains`.

AWS CLI

Para listar domínios em uma lista de domínios

O `list-firewall-domains` exemplo a seguir lista os domínios em uma lista de domínios DNS do Firewall que você especifica.

```
aws route53resolver list-firewall-domains \
```

```
--firewall-domain-list-id rslvr-fdl-d61cbb2cbexample
```

Saída:

```
{
  "Domains": [
    "test1.com.",
    "test2.com.",
    "test3.com."
  ]
}
```

Para obter mais informações, consulte [Gerenciando suas próprias listas de domínios](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListFirewallDomains](#) na Referência de AWS CLI Comandos.

list-firewall-rule-group-associations

O código de exemplo a seguir mostra como usar `list-firewall-rule-group-associations`.

AWS CLI

Para listar associações de grupos de regras do DNS Firewall

O `list-firewall-rule-group-associations` exemplo a seguir lista suas associações de grupos de regras do DNS Firewall com a AmazonVPCs.

```
aws route53resolver list-firewall-rule-group-associations
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Gerenciando associações entre você VPC e o grupo de regras do Route 53 Resolver DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListFirewallRuleGroupAssociations](#) na Referência de AWS CLI Comandos.

list-firewall-rule-groups

O código de exemplo a seguir mostra como usar `list-firewall-rule-groups`.

AWS CLI

Para obter uma lista dos grupos de regras do Firewall

O `list-firewall-rule-groups` exemplo a seguir lista seus grupos de regras DNS do Firewall.

```
aws route53resolver list-firewall-rule-groups
```

Saída:

```
{
  "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"
    }
]
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListFirewallRuleGroups](#) na Referência de AWS CLI Comandos.

list-firewall-rules

O código de exemplo a seguir mostra como usar `list-firewall-rules`.

AWS CLI

Para listar as regras de firewall

O `list-firewall-rules` exemplo a seguir lista todas as suas regras de DNS firewall em um grupo de regras de firewall.

```
aws route53resolver list-firewall-rules \
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

Saída:

```
{
  "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"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListFirewallRules](#) na Referência de AWS CLI Comandos.

list-resolver-endpoint-ip-addresses

O código de exemplo a seguir mostra como usar `list-resolver-endpoint-ip-addresses`.

AWS CLI

Para listar endereços IP para um endpoint de entrada ou saída especificado

O `list-resolver-endpoint-ip-addresses` exemplo a seguir lista informações sobre os endereços IP associados ao endpoint `rslvr-in-f9ab8a03f1example` de entrada. Você também pode usar `list-resolver-endpoint-ip-addresses` para endpoints de saída especificando o ID do endpoint aplicável.

```
aws route53resolver list-resolver-endpoint-ip-addresses \
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações sobre os valores na saída, consulte [Valores que você especifica ao criar ou editar endpoints de entrada](#) e [Valores que você especifica ao criar ou editar endpoints de saída](#), ambos no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [ListResolverEndpointIpAddresses](#) na Referência de AWS CLI Comandos.

list-resolver-endpoints

O código de exemplo a seguir mostra como usar `list-resolver-endpoints`.

AWS CLI

Para listar endpoints do Resolver em uma região AWS

O `list-resolver-endpoints` exemplo a seguir lista os endpoints de entrada e saída do Resolver que existem na conta atual.

```
aws route53resolver list-resolver-endpoints
```

Saída:

```

{
  "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"
    }
]
}

```

- Para API obter detalhes, consulte [ListResolverEndpoints](#) na Referência de AWS CLI Comandos.

list-resolver-rule-associations

O código de exemplo a seguir mostra como usar `list-resolver-rule-associations`.

AWS CLI

Para listar associações entre as regras do Resolver e VPCs

O `list-resolver-rule-associations` exemplo a seguir lista as associações entre as regras do resolvidor e VPCs na AWS conta atual.

```
aws route53resolver list-resolver-rule-associations
```

Saída:

```
{
  "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": ""
    }
  ]
}
```

Para obter mais informações, consulte [Como o Resolver do Route 53 encaminha DNS consultas da sua VPCs para a sua rede](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [ListResolverRuleAssociations](#) na Referência de AWS CLI Comandos.

list-resolver-rules

O código de exemplo a seguir mostra como usar `list-resolver-rules`.

AWS CLI

Para listar as regras do Resolver

O `list-resolver-rules` exemplo a seguir lista todas as regras do Resolver na AWS conta atual.

```
aws route53resolver list-resolver-rules
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Como o Resolver do Route 53 encaminha DNS consultas da sua VPCs para a sua rede](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [ListResolverRules](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso Resolver

O `list-tags-for-resource` exemplo a seguir lista as tags atribuídas à regra Resolver especificada.

```
aws route53resolver list-tags-for-resource \
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
  rslvr-rr-42b60677c0example"
```

Saída:

```
{
  "Tags": [
    {
      "Key": "my-key-1",
      "Value": "my-value-1"
    },
    {
      "Key": "my-key-2",
      "Value": "my-value-2"
    }
  ]
}
```

Para obter informações sobre o uso de tags para alocação de custos, consulte Como [usar tags de alocação de custos](#) no AWS Billing and Cost Management User Guide.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

put-firewall-rule-group-policy

O código de exemplo a seguir mostra como usar `put-firewall-rule-group-policy`.

AWS CLI

Para anexar uma AWS IAM política para compartilhar uma política de grupo de regras de firewall

O `put-firewall-rule-group-policy` exemplo a seguir anexa uma política de AWS Identity and Access Management (AWS IAM) para compartilhar o grupo de regras.

```
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\"}]}"
```

Saída:

```
{
  "ReturnValue": true
}
```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [PutFirewallRuleGroupPolicy](#) na Referência de AWS CLI Comandos.

put-resolver-rule-policy

O código de exemplo a seguir mostra como usar `put-resolver-rule-policy`.

AWS CLI

Para compartilhar uma regra do Resolver com outra AWS conta

O `put-resolver-rule-policy` exemplo a seguir especifica uma regra do Resolver que você deseja compartilhar com outra AWS conta, a conta com a qual você deseja compartilhar a regra e as operações relacionadas à regra que você deseja que a conta possa realizar nas regras.

Observação Você deve executar esse comando usando credenciais da mesma conta que criou a regra.

```
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\" ] } ] }"
```

Saída:

```
{
  "ReturnValue": true
}
```

Depois de executar `put-resolver-rule-policy`, você pode executar os dois comandos do Resource Access Manager (RAM) a seguir. Você deve usar a conta com a qual deseja compartilhar a regra:

`get-resource-share-invitations` retorna o valor `resourceShareInvitationArn`. Você precisa desse valor para aceitar o convite para usar a regra compartilhada. `accept-resource-share-invitation` aceita o convite para usar a regra compartilhada.

Para obter mais informações, consulte a seguinte documentação do :

[get-resource-share-invitationsaccept-resource-share-invitationsCompartilhando regras de encaminhamento com outras AWS contas e usando regras compartilhadas](#) no Guia do desenvolvedor do Amazon Route 53

- Para API obter detalhes, consulte [PutResolverRulePolicy](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para associar tags a um recurso Resolver

O `tag-resource` exemplo a seguir associa dois pares de chave/valor de tag à regra Resolver especificada.

```
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"
```

Este comando não produz saída.

Para obter informações sobre o uso de tags para alocação de custos, consulte Como [usar tags de alocação de custos](#) no AWS Billing and Cost Management User Guide.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um recurso do Resolver

O `untag-resource` exemplo a seguir remove duas tags da regra Resolver especificada.

```
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
```

Este comando não produz saída. Para confirmar que as tags foram removidas, você pode usar [list-tags-for-resource](#).

Para obter informações sobre o uso de tags para alocação de custos, consulte Como [usar tags de alocação de custos](#) no AWS Billing and Cost Management User Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-firewall-config

O código de exemplo a seguir mostra como usar `update-firewall-config`.

AWS CLI

Para atualizar uma configuração de firewall

O `update-firewall-config` exemplo a seguir atualiza a configuração do DNS Firewall.

```
aws route53resolver update-firewall-config \  
  --resource-id vpc-31e92222 \  
  --firewall-fail-open DISABLED
```

Saída:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e92222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

Para obter mais informações, consulte [VPCConfiguração do DNS firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [UpdateFirewallConfig](#) na Referência de AWS CLI Comandos.

update-firewall-domains

O código de exemplo a seguir mostra como usar `update-firewall-domains`.

AWS CLI

Para atualizar uma lista de domínios

O `update-firewall-domains` exemplo a seguir adiciona os domínios a uma lista de domínios com a ID fornecida por você.

```
aws route53resolver update-firewall-domains \  
  --firewall-domain-list-id rslvr-fdl-42b60677cexampleb \  
  --operation ADD \  
  --domains test1.com test2.com test3.com
```

Saída:

```
{  
  "Id": "rslvr-fdl-42b60677cexample",  
  "Name": "test",  
  "Status": "UPDATING",  
  "StatusMessage": "Updating the Firewall Domain List"  
}
```

Para obter mais informações, consulte [Gerenciando suas próprias listas de domínios](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [UpdateFirewallDomains](#) na Referência de AWS CLI Comandos.

update-firewall-rule-group-association

O código de exemplo a seguir mostra como usar update-firewall-rule-group-association.

AWS CLI

Para atualizar uma associação de grupo de regras de firewall

O update-firewall-rule-group-association exemplo a seguir atualiza uma associação de grupo de regras de firewall.

```
aws route53resolver update-firewall-rule-group-association \  
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \  
  --priority 103
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciando associações entre você VPC e o grupo de regras do Route 53 Resolver DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [UpdateFirewallRuleGroupAssociation](#) na Referência de AWS CLI Comandos.

update-firewall-rule

O código de exemplo a seguir mostra como usar `update-firewall-rule`.

AWS CLI

Para atualizar uma regra de firewall

O `update-firewall-rule` exemplo a seguir atualiza uma regra de firewall com os parâmetros que você especifica.

```

aws route53resolver update-firewall-rule \
  --firewall-rule-group-id rslvr-frg-47f93271fexample \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
  --priority 102

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciamento de grupos de regras e regras no DNS Firewall](#) no Amazon Route 53 Developer Guide.

- Para API obter detalhes, consulte [UpdateFirewallRule](#) na Referência de AWS CLI Comandos.

update-resolver-endpoint

O código de exemplo a seguir mostra como usar `update-resolver-endpoint`.

AWS CLI

Para atualizar o nome de um endpoint do Resolver

O `update-resolver-endpoint` exemplo a seguir atualiza o nome de um endpoint do Resolver. A atualização de outros valores não é suportada.

```

aws route53resolver update-resolver-endpoint \
  --resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \
  --name my-renamed-inbound-endpoint

```

Saída:

```

{
  "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"
  }
}

```

- Para API obter detalhes, consulte [UpdateResolverEndpoint](#) na Referência de AWS CLI Comandos.

update-resolver-rule

O código de exemplo a seguir mostra como usar `update-resolver-rule`.

AWS CLI

Exemplo 1: Para atualizar as configurações do endpoint do Resolver

O `update-resolver-rule` exemplo a seguir atualiza o nome da regra, os endereços IP em sua rede local para os quais DNS as consultas são encaminhadas e a ID do endpoint Resolver de saída que você está usando para encaminhar consultas para sua rede.

Observação Os valores existentes para `TargetIps` são substituídos, portanto, você deve especificar todos os endereços IP que deseja que a regra tenha após a atualização.

```

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

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 2: Para atualizar as configurações do endpoint do Resolver usando um arquivo para configurações ``config``

Como alternativa, você pode incluir as config configurações em um JSON arquivo e depois especificar esse arquivo ao `ligarupdate-resolver-rule`.

```

aws route53resolver update-resolver-rule \
  --resolver-rule-id rslvr-rr-1247fa64f3example \
  --config file://c:\temp\update-resolver-rule.json

```

Conteúdo de `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"
}
```

Para obter mais informações, consulte [Valores que você especifica ao criar ou editar regras](#) no Guia do desenvolvedor do Amazon Route 53.

- Para API obter detalhes, consulte [UpdateResolverRule](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon S3 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon S3.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

abort-multipart-upload

O código de exemplo a seguir mostra como usar `abort-multipart-upload`.

AWS CLI

Para interromper um carregamento fracionado especificado

O comando `abort-multipart-upload` a seguir interrompe um carregamento fracionado da chave `multipart/01` no bucket `my-bucket`.

```
aws s3api abort-multipart-upload \
```

```
--bucket my-bucket \  
--key multipart/01 \  
--upload-  
id dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZlJF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

O ID de upload exigido por esse comando é a saída de `create-multipart-upload` e também pode ser recuperado com `list-multipart-uploads`.

- Para API obter detalhes, consulte [AbortMultipartUpload](#) na Referência de AWS CLI Comandos.

complete-multipart-upload

O código de exemplo a seguir mostra como usar `complete-multipart-upload`.

AWS CLI

O seguinte comando conclui um upload fracionado da chave `multipart/01` no bucket `my-bucket`:

```
aws s3api complete-multipart-upload --multipart-upload file://  
mpustruct --bucket my-bucket --key 'multipart/01' --upload-  
id dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZlJF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

O ID de upload exigido por esse comando é a saída de `create-multipart-upload` e também pode ser recuperado com `list-multipart-uploads`.

A opção de upload de várias partes no comando acima usa uma JSON estrutura que descreve as partes do upload de várias partes que devem ser reagrupadas no arquivo completo. Neste exemplo, o `file://` prefixo é usado para carregar a JSON estrutura de um arquivo na pasta local chamada `mpustruct`.

`mpustruct`:

```
{  
  "Parts": [  
    {  
      "ETag": "e868e0f4719e394144ef36531ee6824c",  
      "PartNumber": 1  
    },  
    {  
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
```

```

    "PartNumber": 2
  },
  {
    "ETag": "d0a0112e841abec9c9ec83406f0159c8",
    "PartNumber": 3
  }
]
}

```

O ETag valor de cada parte no upload é gerado toda vez que você carrega uma peça usando o `upload-part` comando e também pode ser recuperado chamando `list-parts` ou calculado usando a MD5 soma de verificação de cada parte.

Saída:

```

{
  "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
  "Bucket": "my-bucket",
  "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
  "Key": "multipart/01"
}

```

- Para API obter detalhes, consulte [CompleteMultipartUpload](#) na Referência de AWS CLI Comandos.

copy-object

O código de exemplo a seguir mostra como usar `copy-object`.

AWS CLI

O comando a seguir copia um objeto de `bucket-1` para `bucket-2`:

```

aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2

```

Saída:

```

{
  "CopyObjectResult": {
    "LastModified": "2015-11-10T01:07:25.000Z",

```

```
    "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
  },
  "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifM1kA"
}
```

- Para API obter detalhes, consulte [CopyObject](#) na Referência de AWS CLI Comandos.

cp

O código de exemplo a seguir mostra como usar cp.

AWS CLI

Exemplo 1: Copiar um arquivo local para o S3

O cp comando a seguir copia um único arquivo para um bucket e uma chave especificados:

```
aws s3 cp test.txt s3://mybucket/test2.txt
```

Saída:

```
upload: test.txt to s3://mybucket/test2.txt
```

Exemplo 2: Copiar um arquivo local para o S3 com uma data de expiração

O cp comando a seguir copia um único arquivo para um bucket e uma chave especificados que expiram no timestamp especificado ISO 8601:

```
aws s3 cp test.txt s3://mybucket/test2.txt \  
  --expires 2014-10-01T20:30:00Z
```

Saída:

```
upload: test.txt to s3://mybucket/test2.txt
```

Exemplo 3: Copiar um arquivo do S3 para o S3

O cp comando a seguir copia um único objeto s3 para um bucket e uma chave especificados:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt
```

Saída:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Exemplo 4: Copiar um objeto do S3 para um arquivo local

O cp comando a seguir copia um único objeto para um arquivo especificado localmente:

```
aws s3 cp s3://mybucket/test.txt test2.txt
```

Saída:

```
download: s3://mybucket/test.txt to test2.txt
```

Exemplo 5: copiar um objeto do S3 de um bucket para outro

O cp comando a seguir copia um único objeto em um bucket especificado, mantendo o nome original:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket2/
```

Saída:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

Exemplo 6: cópia recursiva de objetos do S3 para um diretório local

Quando transmitido com o parâmetro `--recursive`, o cp comando a seguir copia recursivamente todos os objetos com um prefixo e um bucket especificados para um diretório especificado. Neste exemplo, o bucket mybucket tem os objetos test1.txt e test2.txt:

```
aws s3 cp s3://mybucket . \  
--recursive
```

Saída:

```
download: s3://mybucket/test1.txt to test1.txt  
download: s3://mybucket/test2.txt to test2.txt
```

Exemplo 7: cópia recursiva de arquivos locais para o S3

Quando transmitido com o parâmetro `--recursive`, o `cp` comando a seguir copia recursivamente todos os arquivos em um diretório especificado para um bucket e prefixo especificados, enquanto exclui alguns arquivos usando um parâmetro. `--exclude` Neste exemplo, o diretório `myDir` tem os arquivos `test1.txt` e `test2.jpg`:

```
aws s3 cp myDir s3://mybucket/ \  
  --recursive \  
  --exclude "*.jpg"
```

Saída:

```
upload: myDir/test1.txt to s3://mybucket/test1.txt
```

Exemplo 8: cópia recursiva de objetos do S3 para outro bucket

Quando transmitido com o parâmetro `--recursive`, o `cp` comando a seguir copia recursivamente todos os objetos em um bucket especificado para outro bucket, ao mesmo tempo em que exclui alguns objetos usando um `--exclude` parâmetro. Neste exemplo, o bucket `mybucket` tem os objetos `test1.txt` e `another/test1.txt`:

```
aws s3 cp s3://mybucket/ s3://mybucket2/ \  
  --recursive \  
  --exclude "another/*"
```

Saída:

```
copy: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

Você pode combinar `--exclude` `--include` opções para copiar somente objetos que correspondam a um padrão, excluindo todos os outros:

```
aws s3 cp s3://mybucket/logs/ s3://mybucket2/logs/ \  
  --recursive \  
  --exclude "*" \  
  --include "*.log"
```

Saída:

```
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
```

Exemplo 9: Configurando a Lista de Controle de Acesso (ACL) ao copiar um objeto do S3

O `cp` comando a seguir copia um único objeto para um bucket e uma chave especificados enquanto define ACL o `comopublic-read-write`:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt \  
  --acl public-read-write
```

Saída:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Observe que, se você estiver usando a `--acl` opção, certifique-se de que todas IAM as políticas associadas incluam a `"s3:PutObjectAcl"` ação:

```
aws iam get-user-policy \  
  --user-name myuser \  
  --policy-name mypolicy
```

Saída:

```
{  
  "UserName": "myuser",  
  "PolicyName": "mypolicy",  
  "PolicyDocument": {  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Action": [  
          "s3:PutObject",  
          "s3:PutObjectAcl"  
        ],  
        "Resource": [  
          "arn:aws:s3:::mybucket/*"  
        ],  
        "Effect": "Allow",  
        "Sid": "Stmt1234567891234"  
      }  
    ]  
  }  
}
```



```

    }
  ]
}
}
}

```

Exemplo 10: concessão de permissões para um objeto do S3

O `cp` comando a seguir ilustra o uso da `--grants` opção para conceder acesso de leitura a todos os usuários identificados URI e controle total a um usuário específico identificado por sua ID canônica:

```
aws s3 cp file.txt s3://mybucket/ --grants read=uri=http://acs.amazonaws.com/groups/global/AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Saída:

```
upload: file.txt to s3://mybucket/file.txt
```

Exemplo 11: Fazer upload de um fluxo de arquivos local para o S3

PowerShell pode alterar a codificação ou adicionar uma CRLF à entrada canalizada.

O `cp` comando a seguir carrega um fluxo de arquivos local da entrada padrão para um bucket e uma chave especificados:

```
aws s3 cp - s3://mybucket/stream.txt
```

Exemplo 12: Fazer upload de um stream de arquivos local maior que 50 GB para o S3

O `cp` comando a seguir carrega um stream de arquivos local de 51 GB da entrada padrão para um bucket e uma chave especificados. A `--expected-size` opção deve ser fornecida ou o upload poderá falhar quando atingir o limite padrão de 10.000 peças:

```
aws s3 cp - s3://mybucket/stream.txt --expected-size 54760833024
```

Exemplo 13: Fazer o download de um objeto do S3 como um fluxo de arquivos local

PowerShell pode alterar a codificação ou adicionar CRLF a à saída canalizada ou redirecionada.

O `cp` comando a seguir baixa um objeto S3 localmente como um fluxo para a saída padrão. Atualmente, o download como um stream não é compatível com o `--recursive` parâmetro:

```
aws s3 cp s3://mybucket/stream.txt -
```

Exemplo 14: Fazendo upload para um ponto de acesso S3

O `cp` comando a seguir carrega um único arquivo (`mydoc.txt`) para o ponto de acesso (`myaccesspoint`) na chave (`mykey`):

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

Saída:

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

Exemplo 15: Fazendo o download de um ponto de acesso S3

O `cp` comando a seguir baixa um único objeto (`mykey`) do ponto de acesso (`myaccesspoint`) para o arquivo local (`mydoc.txt`):

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey mydoc.txt
```

Saída:

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to mydoc.txt
```

- Para API obter detalhes, consulte [Cp](#) na Referência de AWS CLI Comandos.

create-bucket

O código de exemplo a seguir mostra como usar `create-bucket`.

AWS CLI

Exemplo 1: como criar um bucket

O seguinte exemplo de `create-bucket` cria um bucket chamado `my-bucket`:

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1
```

Saída:

```
{  
  "Location": "/my-bucket"  
}
```

Para obter mais informações, consulte [Como criar um bucket](#) no Guia do usuário do Amazon S3.

Exemplo 2: como criar um bucket com aplicação de políticas de proprietário

O exemplo de `create-bucket` a seguir cria um bucket chamado `my-bucket` que usa a configuração aplicada de proprietário de bucket para a propriedade de objetos do S3.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1 \  
  --object-ownership BucketOwnerEnforced
```

Saída:

```
{  
  "Location": "/my-bucket"  
}
```

Para obter mais informações, consulte [Controle de propriedade de objetos e desativação ACLs](#) no Guia do usuário do Amazon S3.

Exemplo 3: como criar um bucket fora da região ``us-east-1``

O exemplo de `create-bucket` a seguir cria um bucket chamado `my-bucket` na região `eu-west-1`. Regiões fora da `us-east-1` exigem que o `LocationConstraint` adequado seja especificado para poder criar o bucket na região desejada.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region eu-west-1 \  
  --location-constraint EUROPE
```

```
--bucket my-bucket \  
--region eu-west-1 \  
--create-bucket-configuration LocationConstraint=eu-west-1
```

Saída:

```
{  
  "Location": "http://my-bucket.s3.amazonaws.com/"  
}
```

Para obter mais informações, consulte [Como criar um bucket](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [CreateBucket](#) na Referência de AWS CLI Comandos.

create-multipart-upload

O código de exemplo a seguir mostra como usar `create-multipart-upload`.

AWS CLI

O seguinte comando cria um upload fracionado no bucket `my-bucket` com a chave `multipart/01`:

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Saída:

```
{  
  "Bucket": "my-bucket",  
  "UploadId":  
    "dfRtDYU0WwCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3URC",  
  "Key": "multipart/01"  
}
```

O arquivo concluído terá o nome `01` e estará em uma pasta chamada `multipart` no bucket `my-bucket`. Salve o ID de upload, a chave e o nome do bucket para usar com o comando `upload-part`.

- Para API obter detalhes, consulte [CreateMultipartUpload](#) na Referência de AWS CLI Comandos.

delete-bucket-analytics-configuration

O código de exemplo a seguir mostra como usar `delete-bucket-analytics-configuration`.

AWS CLI

Para excluir uma configuração de análise de um bucket

O exemplo `delete-bucket-analytics-configuration` a seguir recupera a configuração referente ao bucket e ID especificados.

```
aws s3api delete-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBucketAnalyticsConfiguration](#) na Referência de AWS CLI Comandos.

delete-bucket-cors

O código de exemplo a seguir mostra como usar `delete-bucket-cors`.

AWS CLI

O seguinte comando exclui a configuração de compartilhamento de recursos de origem cruzada do bucket `my-bucket`:

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- Para API obter detalhes, consulte [DeleteBucketCors](#) na Referência de AWS CLI Comandos.

delete-bucket-encryption

O código de exemplo a seguir mostra como usar `delete-bucket-encryption`.

AWS CLI

Para excluir a configuração de criptografia do lado do servidor de um bucket

O exemplo `delete-bucket-encryption` a seguir exclui a configuração de criptografia do lado do servidor do bucket especificado.

```
aws s3api delete-bucket-encryption \  
  --bucket my-bucket
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBucketEncryption](#) na Referência de AWS CLI Comandos.

delete-bucket-intelligent-tiering-configuration

O código de exemplo a seguir mostra como usar `delete-bucket-intelligent-tiering-configuration`.

AWS CLI

Para remover uma configuração S3 Intelligent-Tiering em um bucket

O `delete-bucket-intelligent-tiering-configuration` exemplo a seguir remove uma configuração S3 Intelligent-Tiering, chamada `ExampleConfig`, em um bucket.

```
aws s3api delete-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id ExampleConfig
```

Este comando não produz saída.

Para obter mais informações, consulte [Usando o S3 Intelligent-Tiering](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [DeleteBucketIntelligentTieringConfiguration](#) na Referência de AWS CLI Comandos.

delete-bucket-inventory-configuration

O código de exemplo a seguir mostra como usar `delete-bucket-inventory-configuration`.

AWS CLI

Para excluir a configuração de inventário de um bucket

O exemplo `delete-bucket-inventory-configuration` a seguir exclui a configuração do inventário com ID 1 do bucket especificado.

```
aws s3api delete-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBucketInventoryConfiguration](#) na Referência de AWS CLI Comandos.

delete-bucket-lifecycle

O código de exemplo a seguir mostra como usar `delete-bucket-lifecycle`.

AWS CLI

O seguinte comando exclui a configuração de ciclo de vida do bucket `my-bucket`:

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- Para API obter detalhes, consulte [DeleteBucketLifecycle](#) na Referência de AWS CLI Comandos.

delete-bucket-metrics-configuration

O código de exemplo a seguir mostra como usar `delete-bucket-metrics-configuration`.

AWS CLI

Para excluir uma configuração de métricas referente a um bucket

O exemplo `delete-bucket-metrics-configuration` a seguir remove a configuração referente ao bucket e ID especificados.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteBucketMetricsConfiguration](#) na Referência de AWS CLI Comandos.

delete-bucket-ownership-controls

O código de exemplo a seguir mostra como usar `delete-bucket-ownership-controls`.

AWS CLI

Para remover as configurações de propriedade de um bucket

O `delete-bucket-ownership-controls` exemplo a seguir remove as configurações de propriedade de um bucket.

```
aws s3api delete-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET
```

Este comando não produz saída.

Para obter mais informações, consulte [Como definir a propriedade do objeto em um bucket existente](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [DeleteBucketOwnershipControls](#) na Referência de AWS CLI Comandos.

delete-bucket-policy

O código de exemplo a seguir mostra como usar `delete-bucket-policy`.

AWS CLI

O seguinte comando exclui uma política de bucket do bucket `my-bucket`:

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- Para API obter detalhes, consulte [DeleteBucketPolicy](#) na Referência de AWS CLI Comandos.

delete-bucket-replication

O código de exemplo a seguir mostra como usar `delete-bucket-replication`.

AWS CLI

O seguinte comando exclui uma configuração de replicação de um bucket denominado my-bucket:

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- Para API obter detalhes, consulte [DeleteBucketReplication](#) na Referência de AWS CLI Comandos.

delete-bucket-tagging

O código de exemplo a seguir mostra como usar delete-bucket-tagging.

AWS CLI

O seguinte comando exclui uma configuração de marcação de um bucket denominado my-bucket:

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- Para API obter detalhes, consulte [DeleteBucketTagging](#) na Referência de AWS CLI Comandos.

delete-bucket-website

O código de exemplo a seguir mostra como usar delete-bucket-website.

AWS CLI

O seguinte comando exclui uma configuração de site do bucket my-bucket:

```
aws s3api delete-bucket-website --bucket my-bucket
```

- Para API obter detalhes, consulte [DeleteBucketWebsite](#) na Referência de AWS CLI Comandos.

delete-bucket

O código de exemplo a seguir mostra como usar delete-bucket.

AWS CLI

O seguinte comando exclui o bucket `my-bucket`:

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- Para API obter detalhes, consulte [DeleteBucket](#) na Referência de AWS CLI Comandos.

`delete-object-tagging`

O código de exemplo a seguir mostra como usar `delete-object-tagging`.

AWS CLI

Para excluir os conjuntos de tags de um objeto

O exemplo `delete-object-tagging` a seguir exclui a tag com a chave especificada do objeto `doc1.rtf`.

```
aws s3api delete-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteObjectTagging](#) na Referência de AWS CLI Comandos.

`delete-object`

O código de exemplo a seguir mostra como usar `delete-object`.

AWS CLI

O seguinte comando exclui um objeto `test.txt` do bucket `my-bucket`:

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Se o versionamento de bucket estiver habilitado, a saída conterá o ID de versão do marcador excluído:

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tND1W1Gq",
  "DeleteMarker": true
}
```

Para obter mais informações sobre como excluir objetos, consulte [Excluir objetos](#) no Guia do desenvolvedor do Amazon S3.

- Para API obter detalhes, consulte [DeleteObject](#) na Referência de AWS CLI Comandos.

delete-objects

O código de exemplo a seguir mostra como usar `delete-objects`.

AWS CLI

O seguinte comando exclui um objeto do bucket `my-bucket`:

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json` é um JSON documento no diretório atual que especifica o objeto a ser excluído:

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Saída:

```
{
  "Deleted": [
    {
      "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUP01koHwzU",
      "Key": "test1.txt",
      "DeleteMarker": true
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [DeleteObjects](#) na Referência de AWS CLI Comandos.

delete-public-access-block

O código de exemplo a seguir mostra como usar `delete-public-access-block`.

AWS CLI

Para excluir a configuração do Bloqueio de Acesso Público para um bucket

O exemplo `delete-public-access-block` a seguir remove a configuração do Bloqueio de Acesso Público no bucket especificado.

```
aws s3api delete-public-access-block \  
  --bucket my-bucket
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePublicAccessBlock](#) na Referência de AWS CLI Comandos.

get-bucket-accelerate-configuration

O código de exemplo a seguir mostra como usar `get-bucket-accelerate-configuration`.

AWS CLI

Para recuperar a configuração de aceleração de um bucket

O exemplo `get-bucket-accelerate-configuration` a seguir recupera a configuração de aceleração do bucket especificado.

```
aws s3api get-bucket-accelerate-configuration \  
  --bucket my-bucket
```

Saída:

```
{
```

```
"Status": "Enabled"
}
```

- Para API obter detalhes, consulte [GetBucketAccelerateConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-acl

O código de exemplo a seguir mostra como usar `get-bucket-acl`.

AWS CLI

O seguinte comando recupera a lista de controle de acesso do bucket `my-bucket`:

```
aws s3api get-bucket-acl --bucket my-bucket
```

Saída:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetBucketAcl](#) na Referência de AWS CLI Comandos.

get-bucket-analytics-configuration

O código de exemplo a seguir mostra como usar `get-bucket-analytics-configuration`.

AWS CLI

Para recuperar a configuração de análise de um bucket com um ID específico

O exemplo `get-bucket-analytics-configuration` a seguir recupera a configuração de análise referente ao bucket e ID especificados.

```
aws s3api get-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Saída:

```
{  
  "AnalyticsConfiguration": {  
    "StorageClassAnalysis": {},  
    "Id": "1"  
  }  
}
```

- Para API obter detalhes, consulte [GetBucketAnalyticsConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-cors

O código de exemplo a seguir mostra como usar `get-bucket-cors`.

AWS CLI

O seguinte comando recupera a configuração de compartilhamento de recursos de origem cruzada para o bucket `my-bucket`:

```
aws s3api get-bucket-cors --bucket my-bucket
```

Saída:

```
{  
  "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": [
      "*"
    ]
  }
]
}

```

- Para API obter detalhes, consulte [GetBucketCors](#) na Referência de AWS CLI Comandos.

get-bucket-encryption

O código de exemplo a seguir mostra como usar `get-bucket-encryption`.

AWS CLI

Para recuperar a configuração de criptografia do lado do servidor para um bucket

O exemplo `get-bucket-encryption` a seguir recupera a configuração de criptografia do lado do servidor do bucket `my-bucket`.

```
aws s3api get-bucket-encryption \
```

```
--bucket my-bucket
```

Saída:

```
{
  "ServerSideEncryptionConfiguration": {
    "Rules": [
      {
        "ApplyServerSideEncryptionByDefault": {
          "SSEAlgorithm": "AES256"
        }
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [GetBucketEncryption](#) na Referência de AWS CLI Comandos.

get-bucket-intelligent-tiering-configuration

O código de exemplo a seguir mostra como usar `get-bucket-intelligent-tiering-configuration`.

AWS CLI

Para recuperar uma configuração S3 Intelligent-Tiering em um bucket

O `get-bucket-intelligent-tiering-configuration` exemplo a seguir recupera uma configuração S3 Intelligent-Tiering, chamada `ExampleConfig`, em um bucket.

```
aws s3api get-bucket-intelligent-tiering-configuration \
  --bucket DOC-EXAMPLE-BUCKET \
  --id ExampleConfig
```

Saída:

```
{
  "IntelligentTieringConfiguration": {
    "Id": "ExampleConfig2",
    "Filter": {
      "Prefix": "images"
    }
  }
}
```



```

    },
    "Status": "Enabled",
    "Tierings": [
      {
        "Days": 90,
        "AccessTier": "ARCHIVE_ACCESS"
      },
      {
        "Days": 180,
        "AccessTier": "DEEP_ARCHIVE_ACCESS"
      }
    ]
  }
}

```

Para obter mais informações, consulte [Usando o S3 Intelligent-Tiering](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [GetBucketIntelligentTieringConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-inventory-configuration

O código de exemplo a seguir mostra como usar `get-bucket-inventory-configuration`.

AWS CLI

Para recuperar a configuração de inventário de um bucket

O exemplo `get-bucket-inventory-configuration` a seguir recupera a configuração de inventário do bucket especificado com ID 1.

```

aws s3api get-bucket-inventory-configuration \
  --bucket my-bucket \
  --id 1

```

Saída:

```

{
  "InventoryConfiguration": {
    "IsEnabled": true,
    "Destination": {

```

```
    "S3BucketDestination": {
      "Format": "ORC",
      "Bucket": "arn:aws:s3:::my-bucket",
      "AccountId": "123456789012"
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
      "Frequency": "Weekly"
    }
  }
}
```

- Para API obter detalhes, consulte [GetBucketInventoryConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-lifecycle-configuration

O código de exemplo a seguir mostra como usar `get-bucket-lifecycle-configuration`.

AWS CLI

O seguinte comando recupera a configuração do ciclo de vida do bucket `my-bucket`:

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Saída:

```
{
  "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"
}
]
```

- Para API obter detalhes, consulte [GetBucketLifecycleConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-lifecycle

O código de exemplo a seguir mostra como usar `get-bucket-lifecycle`.

AWS CLI

O seguinte comando recupera a configuração do ciclo de vida do bucket `my-bucket`:

```
aws s3api get-bucket-lifecycle --bucket my-bucket
```

Saída:

```
{
  "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"
}
]
}
```

- Para API obter detalhes, consulte [GetBucketLifecycle](#) na Referência de AWS CLI Comandos.

get-bucket-location

O código de exemplo a seguir mostra como usar `get-bucket-location`.

AWS CLI

O seguinte comando recupera a restrição de localização do bucket `my-bucket`, se houver uma restrição:

```
aws s3api get-bucket-location --bucket my-bucket
```

Saída:

```
{
  "LocationConstraint": "us-west-2"
}
```

- Para API obter detalhes, consulte [GetBucketLocation](#) na Referência de AWS CLI Comandos.

get-bucket-logging

O código de exemplo a seguir mostra como usar `get-bucket-logging`.

AWS CLI

Para recuperar o status de registro em log de um bucket

O exemplo `get-bucket-logging` a seguir recupera o status de registro em log do bucket especificado.

```
aws s3api get-bucket-logging \  
  --bucket my-bucket
```

Saída:

```
{  
  "LoggingEnabled": {  
    "TargetPrefix": "",  
    "TargetBucket": "my-bucket-logs"  
  }  
}
```

- Para API obter detalhes, consulte [GetBucketLogging](#) na Referência de AWS CLI Comandos.

get-bucket-metrics-configuration

O código de exemplo a seguir mostra como usar `get-bucket-metrics-configuration`.

AWS CLI

Para recuperar a configuração de métricas de um bucket com um ID específico

O exemplo `get-bucket-metrics-configuration` a seguir exibe a configuração de métricas referente ao bucket e ID especificados.

```
aws s3api get-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Saída:

```
{  
  "MetricsConfiguration": {  
    "Filter": {  
      "Prefix": "logs"  
    },  
    "Id": "123"  
  }  
}
```

- Para API obter detalhes, consulte [GetBucketMetricsConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-notification-configuration

O código de exemplo a seguir mostra como usar `get-bucket-notification-configuration`.

AWS CLI

O seguinte comando recupera a configuração de notificação do bucket denominado `my-bucket`:

```
aws s3api get-bucket-notification-configuration --bucket my-bucket
```

Saída:

```
{
  "TopicConfigurations": [
    {
      "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWw1",
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
      "Events": [
        "s3:ObjectCreated:*"
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [GetBucketNotificationConfiguration](#) na Referência de AWS CLI Comandos.

get-bucket-notification

O código de exemplo a seguir mostra como usar `get-bucket-notification`.

AWS CLI

O seguinte comando recupera a configuração de notificação do bucket denominado `my-bucket`:

```
aws s3api get-bucket-notification --bucket my-bucket
```

Saída:

```
{
  "TopicConfiguration": {
    "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
    "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
    "Event": "s3:ObjectCreated:*",
    "Events": [
      "s3:ObjectCreated:*"
    ]
  }
}
```

- Para API obter detalhes, consulte [GetBucketNotification](#) na Referência de AWS CLI Comandos.

get-bucket-ownership-controls

O código de exemplo a seguir mostra como usar `get-bucket-ownership-controls`.

AWS CLI

Para recuperar as configurações de propriedade de um bucket

O `get-bucket-ownership-controls` exemplo a seguir recupera as configurações de propriedade de um bucket.

```
aws s3api get-bucket-ownership-controls \
  --bucket DOC-EXAMPLE-BUCKET
```

Saída:

```
{
  "OwnershipControls": {
    "Rules": [
      {
        "ObjectOwnership": "BucketOwnerEnforced"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Visualização da configuração de propriedade de objetos para um bucket do S3](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [GetBucketOwnershipControls](#) na Referência de AWS CLI Comandos.

get-bucket-policy-status

O código de exemplo a seguir mostra como usar `get-bucket-policy-status`.

AWS CLI

Para recuperar o status da política de um bucket que indica se o bucket é público

O exemplo `get-bucket-policy-status` a seguir recupera o status da política do bucket `my-bucket`.

```
aws s3api get-bucket-policy-status \  
  --bucket my-bucket
```

Saída:

```
{  
  "PolicyStatus": {  
    "IsPublic": false  
  }  
}
```

- Para API obter detalhes, consulte [GetBucketPolicyStatus](#) na Referência de AWS CLI Comandos.

get-bucket-policy

O código de exemplo a seguir mostra como usar `get-bucket-policy`.

AWS CLI

O seguinte comando recupera política de bucket do bucket `my-bucket`:

```
aws s3api get-bucket-policy --bucket my-bucket
```

Saída:


```
{
  "Policy": "{ \"Version\": \"2008-10-17\", \"Statement\": [{ \"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/*\" }, { \"Sid\": \"\", \"Effect\": \"Deny\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/secret/*\" } ] }"
```

Obtenha e coloque um bucket O exemplo a policyThe seguir mostra como você pode baixar uma política de bucket do Amazon S3, fazer modificações no arquivo e depois usá-la put -bucket -policy para aplicar a política de bucket modificada. Para fazer o download de uma política de bucket em um arquivo, você pode executar:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
policy.json
```

Em seguida, modifique o arquivo `policy.json` conforme necessário. Por fim, aplique essa política modificada ao bucket do S3 executando:

o arquivo `policy.json` conforme necessário. Por fim, aplique essa política modificada ao bucket do S3 executando:

conforme necessário. Por fim, aplique essa política modificada ao bucket do S3 executando:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- Para API obter detalhes, consulte [GetBucketPolicy](#) na Referência de AWS CLI Comandos.

get-bucket-replication

O código de exemplo a seguir mostra como usar `get-bucket-replication`.

AWS CLI

O seguinte comando recupera a configuração de replicação de um bucket denominado `my-bucket`:

```
aws s3api get-bucket-replication --bucket my-bucket
```

Saída:

```
{
  "ReplicationConfiguration": {
    "Rules": [
      {
        "Status": "Enabled",
        "Prefix": "",
        "Destination": {
          "Bucket": "arn:aws:s3:::my-bucket-backup",
          "StorageClass": "STANDARD"
        },
        "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
      }
    ],
    "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
  }
}
```

- Para API obter detalhes, consulte [GetBucketReplication](#) na Referência de AWS CLI Comandos.

get-bucket-request-payment

O código de exemplo a seguir mostra como usar `get-bucket-request-payment`.

AWS CLI

Para recuperar a configuração de pagamento de solicitação de um bucket

O exemplo `get-bucket-request-payment` a seguir recupera a configuração de pagamento de solicitação do bucket especificado.

```
aws s3api get-bucket-request-payment \
  --bucket my-bucket
```

Saída:

```
{
  "Payer": "BucketOwner"
}
```

- Para API obter detalhes, consulte [GetBucketRequestPayment](#) na Referência de AWS CLI Comandos.

get-bucket-tagging

O código de exemplo a seguir mostra como usar `get-bucket-tagging`.

AWS CLI

O seguinte comando recupera a configuração de marcação de um bucket denominado `my-bucket`:

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Saída:

```
{
  "TagSet": [
    {
      "Value": "marketing",
      "Key": "organization"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetBucketTagging](#) na Referência de AWS CLI Comandos.

get-bucket-versioning

O código de exemplo a seguir mostra como usar `get-bucket-versioning`.

AWS CLI

O seguinte comando recupera a configuração de versionamento de um bucket denominado `my-bucket`:

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Saída:

```
{
  "Status": "Enabled"
}
```

- Para API obter detalhes, consulte [GetBucketVersioning](#) na Referência de AWS CLI Comandos.

get-bucket-website

O código de exemplo a seguir mostra como usar `get-bucket-website`.

AWS CLI

O seguinte comando recupera a configuração estática do site do bucket `my-bucket`:

```
aws s3api get-bucket-website --bucket my-bucket
```

Saída:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

- Para API obter detalhes, consulte [GetBucketWebsite](#) na Referência de AWS CLI Comandos.

get-object-acl

O código de exemplo a seguir mostra como usar `get-object-acl`.

AWS CLI

O seguinte comando recupera a lista de controle de acesso de um objeto em um bucket chamado `my-bucket`:

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Saída:

```
{
  "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [GetObjectAcl](#) na Referência de AWS CLI Comandos.

get-object-attributes

O código de exemplo a seguir mostra como usar `get-object-attributes`.

AWS CLI

Como recuperar metadados de um objeto sem exibir o próprio objeto

O exemplo de `get-object-attributes` a seguir recupera metadados do objeto `doc1.rtf`.

```

aws s3api get-object-attributes \
  --bucket my-bucket \
  --key doc1.rtf \
  --object-attributes "StorageClass" "ETag" "ObjectSize"

```

Saída:

```

{
  "LastModified": "2022-03-15T19:37:31+00:00",

```

```
"VersionId": "IuCPjXTDzHNf1dAuitVBIKJpF2p1fg4P",
"ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
"StorageClass": "STANDARD",
"ObjectSize": 405
}
```

Para obter mais informações, consulte [GetObjectAttributes](#) na Referência do Amazon S3API.

- Para API obter detalhes, consulte [GetObjectAttributes](#) na Referência de AWS CLI Comandos.

get-object-legal-hold

O código de exemplo a seguir mostra como usar `get-object-legal-hold`.

AWS CLI

Recupera o status de “Retenção jurídica” de um objeto

O exemplo de `get-object-legal-hold` a seguir recupera o status de “Retenção jurídica” do objeto especificado.

```
aws s3api get-object-legal-hold \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf
```

Saída:

```
{
  "LegalHold": {
    "Status": "ON"
  }
}
```

- Para API obter detalhes, consulte [GetObjectLegalHold](#) na Referência de AWS CLI Comandos.

get-object-lock-configuration

O código de exemplo a seguir mostra como usar `get-object-lock-configuration`.

AWS CLI

Como recuperar uma configuração de bloqueio de objetos de um bucket

O exemplo de `get-object-lock-configuration` a seguir recupera a configuração de bloqueio de objetos do bucket especificado.

```
aws s3api get-object-lock-configuration \  
--bucket my-bucket-with-object-lock
```

Saída:

```
{  
  "ObjectLockConfiguration": {  
    "ObjectLockEnabled": "Enabled",  
    "Rule": {  
      "DefaultRetention": {  
        "Mode": "COMPLIANCE",  
        "Days": 50  
      }  
    }  
  }  
}
```

- Para API obter detalhes, consulte [GetObjectLockConfiguration](#) na Referência de AWS CLI Comandos.

get-object-retention

O código de exemplo a seguir mostra como usar `get-object-retention`.

AWS CLI

Como recuperar a configuração de retenção de objetos de um objeto

O exemplo de `get-object-retention` a seguir recupera a configuração de retenção de objetos do objeto especificado.

```
aws s3api get-object-retention \  
--bucket my-bucket-with-object-lock \  
--key doc1.rtf
```

Saída:

```
{
```

```
"Retention": {
  "Mode": "GOVERNANCE",
  "RetainUntilDate": "2025-01-01T00:00:00.000Z"
}
```

- Para API obter detalhes, consulte [GetObjectRetention](#) na Referência de AWS CLI Comandos.

get-object-tagging

O código de exemplo a seguir mostra como usar `get-object-tagging`.

AWS CLI

Para recuperar as tags anexadas a um objeto

O exemplo `get-object-tagging` a seguir recupera os valores da chave especificada do objeto especificado.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc1.rtf
```

Saída:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    }
  ]
}
```

O exemplo `get-object-tagging` a seguir tenta recuperar os conjuntos de tags do objeto `doc2.rtf`, o qual não tem tags.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc2.rtf
```


Saída:

```
{
  "TagSet": []
}
```

O exemplo `get-object-tagging` a seguir recupera os conjuntos de tags do objeto `doc3.rtf`, o qual não tem tags.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc3.rtf
```

Saída:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    },
    {
      "Value": "finance",
      "Key": "department"
    },
    {
      "Value": "payroll",
      "Key": "team"
    }
  ]
}
```

- Para API obter detalhes, consulte [GetObjectTagging](#) na Referência de AWS CLI Comandos.

get-object-torrent

O código de exemplo a seguir mostra como usar `get-object-torrent`.

AWS CLI

O comando a seguir cria um torrent para um objeto em um bucket chamado `my-bucket`:

```
aws s3api get-object-torrent --bucket my-bucket --key large-video-file.mp4 large-video-file.torrent
```

O arquivo torrent é salvo localmente na pasta atual. Observe que o nome do arquivo de saída (*large-video-file.torrent*) é especificado sem um nome de opção e deve ser o último argumento no comando.

- Para API obter detalhes, consulte [GetObjectTorrent](#) na Referência de AWS CLI Comandos.

get-object

O código de exemplo a seguir mostra como usar `get-object`.

AWS CLI

O seguinte exemplo usa o comando `get-object` para fazer download de um objeto do Amazon S3:

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2 my_images.tar.bz2
```

Observe que o parâmetro `outfile` é especificado sem um nome de opção, como “`--outfile`”. O nome do arquivo de saída deve ser o último parâmetro no comando.

O exemplo abaixo demonstra o uso de `--range` para fazer download de um intervalo de bytes específico de um objeto. Observe que os intervalos de bytes precisam ser prefixados com “`bytes=`”:

```
aws s3api get-object --bucket text-content --key dir/my_data --range bytes=8888-9999 my_data_range
```

Para obter mais informações sobre como recuperar objetos, consulte [Obter um objeto](#) no Guia do desenvolvedor do Amazon S3.

- Para API obter detalhes, consulte [GetObject](#) na Referência de AWS CLI Comandos.

get-public-access-block

O código de exemplo a seguir mostra como usar `get-public-access-block`.

AWS CLI

Para definir ou modificar a configuração do Bloqueio de Acesso Público de um bucket

O exemplo `get-public-access-block` a seguir exibe a configuração do Bloqueio de Acesso Público do bucket especificado.

```
aws s3api get-public-access-block \
  --bucket my-bucket
```

Saída:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
    "RestrictPublicBuckets": true
  }
}
```

- Para API obter detalhes, consulte [GetPublicAccessBlock](#) na Referência de AWS CLI Comandos.

head-bucket

O código de exemplo a seguir mostra como usar `head-bucket`.

AWS CLI

O seguinte comando verifica o acesso ao bucket `my-bucket`:

```
aws s3api head-bucket --bucket my-bucket
```

Se o bucket existir e você tiver acesso a ele, nenhuma saída será retornada. Caso contrário, uma mensagem de erro será exibida. Por exemplo:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Para API obter detalhes, consulte [HeadBucket](#) na Referência de AWS CLI Comandos.

head-object

O código de exemplo a seguir mostra como usar `head-object`.

AWS CLI

O seguinte comando recupera metadados de um objeto no bucket `my-bucket`:

```
aws s3api head-object --bucket my-bucket --key index.html
```

Saída:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "text/html",
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
  "ContentLength": 77,
  "VersionId": "null",
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
  "Metadata": {}
}
```

- Para API obter detalhes, consulte [HeadObject](#) na Referência de AWS CLI Comandos.

list-bucket-analytics-configurations

O código de exemplo a seguir mostra como usar `list-bucket-analytics-configurations`.

AWS CLI

Como recuperar uma lista de configurações de análise de um bucket

O exemplo `list-bucket-analytics-configurations` a seguir recupera uma lista de configurações de análise do bucket especificado.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Saída:

```
{
  "AnalyticsConfigurationList": [
```

```
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ],
  "IsTruncated": false
}
```

- Para API obter detalhes, consulte [ListBucketAnalyticsConfigurations](#) na Referência de AWS CLI Comandos.

list-bucket-intelligent-tiering-configurations

O código de exemplo a seguir mostra como usar `list-bucket-intelligent-tiering-configurations`.

AWS CLI

Para recuperar todas as configurações do S3 Intelligent-Tiering em um bucket

O `list-bucket-intelligent-tiering-configurations` exemplo a seguir recupera toda a configuração do S3 Intelligent-Tiering em um bucket.

```
aws s3api list-bucket-intelligent-tiering-configurations \
  --bucket DOC-EXAMPLE-BUCKET
```

Saída:

```
{
  "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"
            }
        ]
    }
]
```

Para obter mais informações, consulte [Usando o S3 Intelligent-Tiering](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [ListBucketIntelligentTieringConfigurations](#) na Referência de AWS CLI Comandos.

list-bucket-inventory-configurations

O código de exemplo a seguir mostra como usar `list-bucket-inventory-configurations`.

AWS CLI

Como recuperar uma lista de configurações de inventário de um bucket

O exemplo `list-bucket-inventory-configurations` a seguir recupera as configurações de inventário do bucket especificado.

```
aws s3api list-bucket-inventory-configurations \
  --bucket my-bucket
```

Saída:

```
{
  "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
}
```

- Para API obter detalhes, consulte [ListBucketInventoryConfigurations](#) na Referência de AWS CLI Comandos.

list-bucket-metrics-configurations

O código de exemplo a seguir mostra como usar `list-bucket-metrics-configurations`.

AWS CLI

Para recuperar uma lista de configurações de métricas para um bucket

O `list-bucket-metrics-configurations` exemplo a seguir recupera uma lista de configurações de métricas para o bucket especificado.

```
aws s3api list-bucket-metrics-configurations \
  --bucket my-bucket
```

Saída:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```



```
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListBucketMetricsConfigurations](#) na Referência de AWS CLI Comandos.

list-buckets

O código de exemplo a seguir mostra como usar `list-buckets`.

AWS CLI

O seguinte comando usa o comando `list-buckets` para exibir os nomes de todos os buckets do Amazon S3 (em todas as regiões):

```
aws s3api list-buckets --query "Buckets[].Name"
```

A opção de consulta filtra a saída de `list-buckets` até sobrar apenas os nomes dos buckets.

Para obter mais informações sobre buckets, consulte *Working with Amazon S3 Buckets* no Guia do desenvolvedor do Amazon S3.

- Para API obter detalhes, consulte [ListBuckets](#) na Referência de AWS CLI Comandos.

list-multipart-uploads

O código de exemplo a seguir mostra como usar `list-multipart-uploads`.

AWS CLI

O seguinte comando lista todos os uploads fracionados ativos do bucket `my-bucket`:

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Saída:

```
{  
  "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": []
}

```

Os uploads fracionados em andamento incorrem em custos de armazenamento no Amazon S3. Conclua ou cancele um upload fracionado ativo para remover suas partes da conta.

- Para API obter detalhes, consulte [ListMultipartUploads](#) na Referência de AWS CLI Comandos.

list-object-versions

O código de exemplo a seguir mostra como usar `list-object-versions`.

AWS CLI

O seguinte comando recupera as informações da versão de um objeto do bucket `my-bucket`:

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Saída:

```

{
  "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
    }
  ]
}

```

- Para API obter detalhes, consulte [ListObjectVersions](#) na Referência de AWS CLI Comandos.

list-objects-v2

O código de exemplo a seguir mostra como usar `list-objects-v2`.

AWS CLI

Como obter uma lista dos objetos em um bucket

O exemplo de `list-objects-v2` a seguir lista os objetos no bucket especificado.

```

aws s3api list-objects-v2 \
  --bucket my-bucket

```

Saída:

```

{
  "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
    }
]
}

```

- Para API obter detalhes, consulte [ListObjectsV2](#) na Referência de AWS CLI Comandos.

list-objects

O código de exemplo a seguir mostra como usar `list-objects`.

AWS CLI

O seguinte exemplo usa o comando `list-objects` para exibir os nomes de todos os objetos presentes no bucket especificado:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

O exemplo usa o argumento `--query` para filtrar a saída de `list-objects` até encontrar o valor e o tamanho da chave de cada objeto

Para obter mais informações sobre objetos, consulte [Working with Amazon S3 Buckets](#) no Guia do desenvolvedor do Amazon S3.

- Para API obter detalhes, consulte [ListObjects](#) na Referência de AWS CLI Comandos.

list-parts

O código de exemplo a seguir mostra como usar `list-parts`.

AWS CLI

O comando a seguir lista todas as partes que foram carregadas para um upload de várias partes com a chave `multipart/01` no bucket `my-bucket`:

```
aws s3api list-parts --bucket my-bucket --key 'multipart/01' --upload-id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [ListParts](#) na Referência de AWS CLI Comandos.

ls

O código de exemplo a seguir mostra como usar `ls`.

AWS CLI

Exemplo 1: listagem de todos os buckets de propriedade do usuário

O `ls` comando a seguir lista todo o bucket de propriedade do usuário. Neste exemplo, o usuário é dono dos buckets `mybucket` e `mybucket2`. O carimbo de data/hora é a data em que o bucket foi criado, mostrada no fuso horário da sua máquina. Essa data pode mudar ao fazer alterações em seu bucket, como editar sua política de bucket. Observe que se `s3://` for usado para o argumento `path<S3Uri>`, ele também listará todos os buckets.

```
aws s3 ls
```

Saída:

```
2013-07-11 17:08:50 mybucket
2013-07-24 14:55:44 mybucket2
```

Exemplo 2: listagem de todos os prefixos e objetos em um bucket

O `ls` comando a seguir lista objetos e prefixos comuns em um bucket e prefixo especificados. Neste exemplo, o usuário é dono do bucket `mybucket` com os objetos `test.txt` `somePrefix/test.txt` e `Os`. `LastWriteTime` e `Length` são arbitrários. Observe que, como o `ls` comando não tem interação com o sistema de arquivos local, o `s3://` URI esquema não é necessário para resolver a ambigüidade e pode ser omitido.

```
aws s3 ls s3://mybucket
```

Saída:

```
2013-07-25 17:06:27      PRE somePrefix/
                        88 test.txt
```

Exemplo 3: listagem de todos os prefixos e objetos em um bucket e prefixo específicos

O `ls` comando a seguir lista objetos e prefixos comuns em um bucket e prefixo especificados. No entanto, não há objetos nem prefixos comuns no bucket e no prefixo especificados.

```
aws s3 ls s3://mybucket/noExistPrefix
```

Saída:

```
None
```

Exemplo 4: listagem recursiva de todos os prefixos e objetos em um bucket

O `ls` comando a seguir listará recursivamente os objetos em um bucket. Em vez de aparecer `PRE dirname/` na saída, todo o conteúdo em um bucket será listado em ordem.

```
aws s3 ls s3://mybucket \
--recursive
```

Saída:

```
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
```

Exemplo 5: resumindo todos os prefixos e objetos em um bucket

O `ls` comando a seguir demonstra o mesmo comando usando as opções `--human-readable` e `--summarize`. `--human-readable` exibe o tamanho do arquivo em bytes/mib/kib/gib/TIB/PIB/EIB. `--summarize` exibe o número total de objetos e o tamanho total no final da lista de resultados:

```
aws s3 ls s3://mybucket \  
  --recursive \  
  --human-readable \  
  --summarize
```

Saída:

```
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
```

Exemplo 6: listagem a partir de um ponto de acesso S3

O `ls` comando a seguir lista objetos do ponto de acesso (myaccesspoint):

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Saída:

```
                PRE somePrefix/  
2013-07-25 17:06:27      88 test.txt
```

- Para API obter detalhes, consulte [Ls](#) na Referência de AWS CLI Comandos.

mb

O código de exemplo a seguir mostra como usar `mb`.

AWS CLI

Exemplo 1: criar um bucket

O `mb` comando a seguir cria um bucket. Neste exemplo, o usuário cria o bucket `mybucket`. O bucket é criado na região especificada no arquivo de configuração do usuário:

```
aws s3 mb s3://mybucket
```

Saída:

```
make_bucket: s3://mybucket
```

Exemplo 2: criar um bucket na região especificada

O `mb` comando a seguir cria um bucket em uma região especificada pelo `--region` parâmetro. Neste exemplo, o usuário cria o bucket `mybucket` na região `us-west-1`:

```
aws s3 mb s3://mybucket \  
--region us-west-1
```

Saída:

```
make_bucket: s3://mybucket
```

- Para API obter detalhes, consulte [Mb](#) na Referência de AWS CLI Comandos.

mv

O código de exemplo a seguir mostra como usar `mv`.

AWS CLI

Exemplo 1: mover um arquivo local para o bucket especificado

O `mv` comando a seguir move um único arquivo para um bucket e uma chave especificados.

```
aws s3 mv test.txt s3://mybucket/test2.txt
```

Saída:

```
move: test.txt to s3://mybucket/test2.txt
```

Exemplo 2: mover um objeto para o bucket e a chave especificados

O mv comando a seguir move um único objeto s3 para um bucket e uma chave especificados.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt
```

Saída:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Exemplo 3: mover um objeto do S3 para o diretório local

O mv comando a seguir move um único objeto para um arquivo especificado localmente.

```
aws s3 mv s3://mybucket/test.txt test2.txt
```

Saída:

```
move: s3://mybucket/test.txt to test2.txt
```

Exemplo 4: mover um objeto com seu nome original para o bucket especificado

O mv comando a seguir move um único objeto para um bucket especificado, mantendo o nome original:

```
aws s3 mv s3://mybucket/test.txt s3://mybucket2/
```

Saída:

```
move: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

Exemplo 5: mover todos os objetos e prefixos em um bucket para o diretório local

Quando transmitido com o parâmetro `--recursive`, o mv comando a seguir move recursivamente todos os objetos com um prefixo e um bucket especificados para um diretório especificado. Neste exemplo, o bucket mybucket tem os objetos test1.txt test2.txt e.

```
aws s3 mv s3://mybucket . \
--recursive
```

Saída:

```
move: s3://mybucket/test1.txt to test1.txt
move: s3://mybucket/test2.txt to test2.txt
```

Exemplo 6: Mova todos os objetos e prefixos em um bucket para o diretório local, exceto os arquivos ``.jpg``

Quando transmitido com o parâmetro `--recursive`, o `mv` comando a seguir move recursivamente todos os arquivos em um diretório especificado para um bucket e prefixo especificados, enquanto exclui alguns arquivos usando um parâmetro. `--exclude` Neste exemplo, o diretório `myDir` tem os arquivos `test1.txt` `test2.jpg` e.

```
aws s3 mv myDir s3://mybucket/ \
--recursive \
--exclude "*.jpg"
```

Saída:

```
move: myDir/test1.txt to s3://mybucket2/test1.txt
```

Exemplo 7: mover todos os objetos e prefixos em um bucket para o diretório local, exceto o prefixo especificado

Quando transmitido com o parâmetro `--recursive`, o `mv` comando a seguir move recursivamente todos os objetos em um intervalo especificado para outro, ao mesmo tempo em que exclui alguns objetos usando um `--exclude` parâmetro. Neste exemplo, o bucket `mybucket` tem os objetos `test1.txt` `another/test1.txt` e.

```
aws s3 mv s3://mybucket/ s3://mybucket2/ \
--recursive \
--exclude "mybucket/another/*"
```

Saída:

```
move: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

Exemplo 8: mover um objeto para o bucket especificado e definir o ACL

O `mv` comando a seguir move um único objeto para um bucket e uma chave especificados ACL ao definir `public-read-write` o.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt \  
--acl public-read-write
```

Saída:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Exemplo 9: mover um arquivo local para o bucket especificado e conceder permissões

O `mv` comando a seguir ilustra o uso da `--grants` opção para conceder acesso de leitura a todos os usuários e controle total a um usuário específico identificado pelo endereço de e-mail.

```
aws s3 mv file.txt s3://mybucket/ \  
--grants read=uri=http://acs.amazonaws.com/groups/global/  
AllUsers full=emailaddress=user@example.com
```

Saída:

```
move: file.txt to s3://mybucket/file.txt
```

Exemplo 10: mover um arquivo para um ponto de acesso do S3

O `mv` comando a seguir move um único arquivo nomeado `mydoc.txt` para o ponto de acesso nomeado `myaccesspoint` na chave nomeada `mykey`.

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

Saída:

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/  
mykey
```

- Para API obter detalhes, consulte [Mv](#) na Referência de AWS CLI Comandos.

presign

O código de exemplo a seguir mostra como usar presign.

AWS CLI

Exemplo 1: Para criar um pré-assinado URL com a vida útil padrão de uma hora vinculado a um objeto em um bucket do S3

O presign comando a seguir gera um pré-assinado URL para um bucket e uma chave especificados que são válidos por uma hora.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt
```

Saída:

```
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
```

Exemplo 2: Para criar um pré-assinado URL com uma vida útil personalizada vinculada a um objeto em um bucket do S3

O presign comando a seguir gera um pré-assinado URL para um bucket e uma chave especificados que são válidos por uma semana.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt \
  --expires-in 604800
```

Saída:

```
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
```

Para obter mais informações, consulte [Compartilhar um objeto com outras pessoas](#) no guia do desenvolvedor do S3.

- Para API obter detalhes, consulte [Presign](#) in AWS CLI Command Reference.

put-bucket-accelerate-configuration

O código de exemplo a seguir mostra como usar `put-bucket-accelerate-configuration`.

AWS CLI

Para definir a configuração de aceleração de um bucket

O exemplo `put-bucket-accelerate-configuration` a seguir habilita a configuração de aceleração do bucket especificado.

```
aws s3api put-bucket-accelerate-configuration \  
  --bucket my-bucket \  
  --accelerate-configuration Status=Enabled
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketAccelerateConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-acl

O código de exemplo a seguir mostra como usar `put-bucket-acl`.

AWS CLI

Este exemplo `full control` concede `read` permissão a dois AWS usuários (`user1@example.com` e `user2@example.com`) a todos:

```
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
```

Consulte <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> para obter detalhes sobre personalização ACLs (os ACL comandos `s3api`, por exemplo `put-bucket-acl`, usam a mesma notação abreviada de argumentos).

- Para API obter detalhes, consulte [PutBucketAcl](#) na Referência de AWS CLI Comandos.

put-bucket-analytics-configuration

O código de exemplo a seguir mostra como usar `put-bucket-analytics-configuration`.

AWS CLI

Para definir uma configuração de análise para o bucket

O `put-bucket-analytics-configuration` exemplo a seguir configura a análise para o bucket especificado.

```
aws s3api put-bucket-analytics-configuration \  
  --bucket my-bucket --id 1 \  
  --analytics-configuration '{"Id": "1", "StorageClassAnalysis": {}}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketAnalyticsConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-cors

O código de exemplo a seguir mostra como usar `put-bucket-cors`.

AWS CLI

O seguinte exemplo habilita solicitações PUT, POST e DELETE de `www.example.com` e solicitações GET de qualquer domínio:

```
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
}
]
}
```

- Para API obter detalhes, consulte [PutBucketCors](#) na Referência de AWS CLI Comandos.

put-bucket-encryption

O código de exemplo a seguir mostra como usar `put-bucket-encryption`.

AWS CLI

Para configurar a criptografia do lado do servidor para um bucket

O `put-bucket-encryption` exemplo a seguir define a AES256 criptografia como padrão para o bucket especificado.

```
aws s3api put-bucket-encryption \
  --bucket my-bucket \
  --server-side-encryption-configuration '{"Rules":
  [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketEncryption](#) na Referência de AWS CLI Comandos.

put-bucket-intelligent-tiering-configuration

O código de exemplo a seguir mostra como usar `put-bucket-intelligent-tiering-configuration`.

AWS CLI

Para atualizar uma configuração do S3 Intelligent-Tiering em um bucket

O `put-bucket-intelligent-tiering-configuration` exemplo a seguir atualiza uma configuração do S3 Intelligent-Tiering, chamada `ExampleConfig`, em um bucket. A configuração fará a transição de objetos que não foram acessados com o prefixo `images` para `Archive Access` após 90 dias e `Deep Archive Access` após 180 dias.

```
aws s3api put-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id "ExampleConfig" \  
  --intelligent-tiering-configuration file://intelligent-tiering-  
configuration.json
```

Conteúdo de `intelligent-tiering-configuration.json`:

```
{  
  "Id": "ExampleConfig",  
  "Status": "Enabled",  
  "Filter": {  
    "Prefix": "images"  
  },  
  "Tierings": [  
    {  
      "Days": 90,  
      "AccessTier": "ARCHIVE_ACCESS"  
    },  
    {  
      "Days": 180,  
      "AccessTier": "DEEP_ARCHIVE_ACCESS"  
    }  
  ]  
}
```

Este comando não produz saída.

Para obter mais informações, consulte [Como definir a propriedade do objeto em um bucket existente](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [PutBucketIntelligentTieringConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-inventory-configuration

O código de exemplo a seguir mostra como usar `put-bucket-inventory-configuration`.

AWS CLI

Exemplo 1: Para definir uma configuração de inventário para um bucket

O `put-bucket-inventory-configuration` exemplo a seguir define um relatório ORC de inventário semanal formatado para o 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" } }'
```

Este comando não produz saída.

Exemplo 2: Para definir uma configuração de inventário para um bucket

O `put-bucket-inventory-configuration` exemplo a seguir define um relatório CSV de inventário formatado diariamente para o 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" } }'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketInventoryConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-lifecycle-configuration

O código de exemplo a seguir mostra como usar `put-bucket-lifecycle-configuration`.

AWS CLI

O seguinte comando aplica a configuração de ciclo de vida ao bucket `my-bucket`:

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-  
configuration file:///lifecycle.json
```

O arquivo `lifecycle.json` é um JSON documento na pasta atual que especifica duas regras:

```
{
  "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"
    }
  ]
}
```

A primeira regra move arquivos com o prefixo `rotated` para o Glacier na data especificada. A segunda regra move versões antigas de objetos para o Glacier quando elas não estão mais atualizadas. Para obter informações sobre formatos de carimbo de data/hora aceitáveis, consulte [Especificação de valores de parâmetros no Guia do AWS CLI usuário](#).

- Para API obter detalhes, consulte [PutBucketLifecycleConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-lifecycle

O código de exemplo a seguir mostra como usar `put-bucket-lifecycle`.

AWS CLI

O comando a seguir aplica uma configuração de ciclo de vida ao bucket: `my-bucket`

```
aws s3api put-bucket-lifecycle --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

O arquivo `lifecycle.json` é um JSON documento na pasta atual que especifica duas regras:

```
{
  "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"
    }
  ]
}
```

A primeira regra move os arquivos para o Amazon Glacier após sessenta dias. A segunda regra exclui arquivos do Amazon S3 na data especificada. Para obter informações sobre formatos de carimbo de data/hora aceitáveis, consulte Especificação de valores de parâmetros no Guia do AWS CLI usuário.

Cada regra no exemplo acima especifica uma política (`Transition` ou `Expiration`) e um prefixo de arquivo (nome da pasta) aos quais ela se aplica. Você também pode criar uma regra que se aplique a um bucket inteiro especificando um prefixo em branco:

```
{
```

```

"Rules": [
  {
    "ID": "Move to Glacier after sixty days (all objects in bucket)",
    "Prefix": "",
    "Status": "Enabled",
    "Transition": {
      "Days": 60,
      "StorageClass": "GLACIER"
    }
  }
]
}

```

- Para API obter detalhes, consulte [PutBucketLifecycle](#) na Referência de AWS CLI Comandos.

put-bucket-logging

O código de exemplo a seguir mostra como usar `put-bucket-logging`.

AWS CLI

Exemplo 1: definir o registro em log da política de bucket

O `put-bucket-logging` exemplo a seguir define a política de registro para `MyBucket`. Primeiro, conceda permissão à entidade principal do serviço de registro em log na política de bucket usando o comando `put-bucket-policy`.

```

aws s3api put-bucket-policy \
  --bucket MyBucket \
  --policy file://policy.json

```

Conteúdo de `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"}
        }
    ]
}

```

Para aplicar a política de registro em log, use `put-bucket-logging`.

```

aws s3api put-bucket-logging \
  --bucket MyBucket \
  --bucket-logging-status file://logging.json

```

Conteúdo de `logging.json`:

```

{
  "LoggingEnabled": {
    "TargetBucket": "MyBucket",
    "TargetPrefix": "Logs/"
  }
}

```

O comando `put-bucket-policy` é necessário para conceder as permissões `s3:PutObject` à entidade principal do serviço de registro em log.

Consulte mais informações em [Registrar em log as solicitações com registro em log de acesso ao servidor](#) no Guia do usuário do Amazon S3.

Exemplo 2: definir uma política de bucket para registrar em log o acesso a um único usuário

O `put-bucket-logging` exemplo a seguir define a política de registro para `MyBucket`. O AWS usuário `bob@example.com` terá controle total sobre os arquivos de log e ninguém mais terá acesso. Primeiro, conceda permissão ao S3 com `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

```

Depois, aplique a política de registro em log usando `put-bucket-logging`.

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file://logging.json
```

Conteúdo de logging.json:

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "MyBucketLogs/",  
    "TargetGrants": [  
      {  
        "Grantee": {  
          "Type": "AmazonCustomerByEmail",  
          "EmailAddress": "bob@example.com"  
        },  
        "Permission": "FULL_CONTROL"  
      }  
    ]  
  }  
}
```

O comando `put-bucket-acl` é necessário para conceder as permissões necessárias ao sistema de entrega de log do S3 (permissões de gravação e `read-acp`).

Consulte mais informações em [Habilitar o log de acesso ao servidor do Amazon S3](#) no Guia do desenvolvedor do Amazon S3.

- Para API obter detalhes, consulte [PutBucketLogging](#) na Referência de AWS CLI Comandos.

put-bucket-metrics-configuration

O código de exemplo a seguir mostra como usar `put-bucket-metrics-configuration`.

AWS CLI

Para definir uma configuração de métricas para um bucket

O `put-bucket-metrics-configuration` exemplo a seguir define uma configuração métrica com ID 123 para o bucket especificado.

```
aws s3api put-bucket-metrics-configuration \  
  --bucket MyBucket \  
  --metric-id 123 \  
  --metric-name MyMetric \  
  --metric-prefix MyMetricPrefix \  
  --metric-unit MyMetricUnit \  
  --metric-type MyMetricType \  
  --metric-status MyMetricStatus \  
  --metric-grants MyMetricGrants \  
  --metric-configuration MyMetricConfiguration
```



```
--bucket my-bucket \  
--id 123 \  
--metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketMetricsConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-notification-configuration

O código de exemplo a seguir mostra como usar `put-bucket-notification-configuration`.

AWS CLI

Como habilitar as notificações especificadas em um bucket

O exemplo de `put-bucket-notification-configuration` a seguir aplica uma configuração de notificação a um bucket chamado `my-bucket`. O arquivo `notification.json` é um JSON documento na pasta atual que especifica um SNS tópico e um tipo de evento a ser monitorado.

```
aws s3api put-bucket-notification-configuration \  
--bucket my-bucket \  
--notification-configuration file://notification.json
```

Conteúdo de `notification.json`:

```
{  
  "TopicConfigurations": [  
    {  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",  
      "Events": [  
        "s3:ObjectCreated:*"  
      ]  
    }  
  ]  
}
```

O SNS tópico deve ter uma IAM política anexada que permita que o Amazon S3 publique nele.

```
{
```

```

"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"
      }
    }
  }
]
}

```

- Para API obter detalhes, consulte [PutBucketNotificationConfiguration](#) na Referência de AWS CLI Comandos.

put-bucket-notification

O código de exemplo a seguir mostra como usar `put-bucket-notification`.

AWS CLI

O exemplo aplica uma configuração de notificação ao bucket `my-bucket`:

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration file://notification.json
```

O arquivo `notification.json` é um JSON documento na pasta atual que especifica um SNS tópico e um tipo de evento para monitorar:

```
{
  "TopicConfiguration": {
```

```
"Event": "s3:ObjectCreated:*",
"Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
}
}
```

O SNS tópico deve ter uma IAM política anexada que permita que o Amazon S3 publique nele:

```
{
  "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"
        }
      }
    }
  ]
}
```

- Para API obter detalhes, consulte [PutBucketNotification](#) na Referência de AWS CLI Comandos.

put-bucket-ownership-controls

O código de exemplo a seguir mostra como usar `put-bucket-ownership-controls`.

AWS CLI

Para atualizar as configurações de propriedade de um bucket

O `put-bucket-ownership-controls` exemplo a seguir atualiza as configurações de propriedade de um bucket.

```
aws s3api put-bucket-ownership-controls \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Este comando não produz saída.

Para obter mais informações, consulte [Como definir a propriedade do objeto em um bucket existente](#) no Guia do usuário do Amazon S3.

- Para API obter detalhes, consulte [PutBucketOwnershipControls](#) na Referência de AWS CLI Comandos.

put-bucket-policy

O código de exemplo a seguir mostra como usar `put-bucket-policy`.

AWS CLI

Este exemplo permite que todos os usuários recuperem qualquer objeto MyBucket, exceto aqueles no MySecretFolder. Também `put` concede `delete` permissão ao usuário raiz da AWS conta1234-5678-9012:

```
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/*"
}
]
}

```

- Para API obter detalhes, consulte [PutBucketPolicy](#) na Referência de AWS CLI Comandos.

put-bucket-replication

O código de exemplo a seguir mostra como usar `put-bucket-replication`.

AWS CLI

Como configurar a replicação para um bucket do S3

O exemplo `put-bucket-replication` a seguir aplica uma configuração de replicação ao bucket do S3 especificado.

```

aws s3api put-bucket-replication \
  --bucket AWSDOC-EXAMPLE-BUCKET1 \
  --replication-configuration file://replication.json

```

Conteúdo de `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"
      }
    }
  ]
}

```

```

    }
  ]
}

```

O bucket de destino deve ter o versionamento habilitado. A função especificada deve ter permissão para gravar no bucket de destino e ter uma relação de confiança que permita que o Amazon S3 assuma a função.

Exemplo de política de permissão de função:

```

{
  "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/*"
    }
  ]
}

```

```
]
}
```

Exemplo de política de relação de confiança:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Este comando não produz saída.

Para obter mais informações, consulte [This is the topic title](#) no Guia do usuário do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [PutBucketReplication](#) na Referência de AWS CLI Comandos.

put-bucket-request-payment

O código de exemplo a seguir mostra como usar `put-bucket-request-payment`.

AWS CLI

Exemplo 1: para habilitar a configuração “requester pays” para um bucket

O exemplo `put-bucket-request-payment` a seguir habilita `requester pays` do bucket especificado.

```
aws s3api put-bucket-request-payment \
  --bucket my-bucket \
  --request-payment-configuration '{"Payer": "Requester"}'
```

Este comando não produz saída.

Exemplo 2: para desabilitar a configuração “requester pays” para um bucket

O exemplo `put-bucket-request-payment` a seguir desabilita `requester pays` do bucket especificado.

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutBucketRequestPayment](#) na Referência de AWS CLI Comandos.

put-bucket-tagging

O código de exemplo a seguir mostra como usar `put-bucket-tagging`.

AWS CLI

O seguinte comando aplica a configuração de marcação a um bucket denominado `my-bucket`:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

O arquivo `tagging.json` é um JSON documento na pasta atual que especifica as tags:

```
{  
  "TagSet": [  
    {  
      "Key": "organization",  
      "Value": "marketing"  
    }  
  ]  
}
```

Ou aplique uma configuração de marcação a `my-bucket` diretamente da linha de comando:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging  
  'TagSet=[{Key=organization,Value=marketing}]'
```


- Para API obter detalhes, consulte [PutBucketTagging](#) na Referência de AWS CLI Comandos.

put-bucket-versioning

O código de exemplo a seguir mostra como usar `put-bucket-versioning`.

AWS CLI

O comando a seguir habilita o versionamento em um bucket denominado `my-bucket`.

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled
```

O comando a seguir habilita o versionamento e usa um código mfa.

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration Status=Enabled --mfa "SERIAL 123456"
```

- Para API obter detalhes, consulte [PutBucketVersioning](#) na Referência de AWS CLI Comandos.

put-bucket-website

O código de exemplo a seguir mostra como usar `put-bucket-website`.

AWS CLI

Aplica uma configuração de site estático ao bucket `my-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://website.json
```

O arquivo `website.json` é um JSON documento na pasta atual que especifica as páginas de índice e erro do site:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
  "ErrorDocument": {
    "Key": "error.html"
  }
}
```

```
}  
}
```

- Para API obter detalhes, consulte [PutBucketWebsite](#) na Referência de AWS CLI Comandos.

put-object-acl

O código de exemplo a seguir mostra como usar `put-object-acl`.

AWS CLI

O comando a seguir concede `read` permissão `full control` a dois AWS usuários (`user1@example.com` e `user2@example.com`) a todos:

```
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
```

Consulte <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> para obter detalhes sobre personalização ACLs (os ACL comandos `s3api`, por exemplo `put-object-acl`, usam a mesma notação abreviada de argumentos).

- Para API obter detalhes, consulte [PutObjectAcl](#) na Referência de AWS CLI Comandos.

put-object-legal-hold

O código de exemplo a seguir mostra como usar `put-object-legal-hold`.

AWS CLI

Como aplicar uma retenção legal a um objeto

O exemplo de `put-object-legal-hold` a seguir define uma retenção legal no objeto `doc1.rtf`.

```
aws s3api put-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --legal-hold Status=ON
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutObjectLegalHold](#) na Referência de AWS CLI Comandos.

put-object-lock-configuration

O código de exemplo a seguir mostra como usar `put-object-lock-configuration`.

AWS CLI

Como definir a configuração de bloqueio de objetos em um bucket

O exemplo de `put-object-lock-configuration` a seguir define um bloqueio de objetos de 50 dias no bucket especificado.

```
aws s3api put-object-lock-configuration \  
  --bucket my-bucket-with-object-lock \  
  --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":  
  { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutObjectLockConfiguration](#) na Referência de AWS CLI Comandos.

put-object-retention

O código de exemplo a seguir mostra como usar `put-object-retention`.

AWS CLI

Como definir uma configuração de retenção de objetos para um objeto

O exemplo de `put-object-retention` a seguir define uma configuração de retenção de objetos para o objeto especificado até 2025-01-01.

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutObjectRetention](#) na Referência de AWS CLI Comandos.

put-object-tagging

O código de exemplo a seguir mostra como usar `put-object-tagging`.

AWS CLI

Para definir uma tag em um objeto

O `put-object-tagging` exemplo a seguir define uma tag com a chave `designation` e o valor `confidential` no objeto especificado.

```
aws s3api put-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf \  
  --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" } ]}'
```

Este comando não produz saída.

O `put-object-tagging` exemplo a seguir define vários conjuntos de tags no objeto especificado.

```
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" } ]}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutObjectTagging](#) na Referência de AWS CLI Comandos.

put-object

O código de exemplo a seguir mostra como usar `put-object`.

AWS CLI

O seguinte exemplo usa o comando `put-object` para fazer upload de um objeto no Amazon S3:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --  
body my_images.tar.bz2
```

O seguinte exemplo mostra o upload de um arquivo de vídeo (o arquivo de vídeo é especificado usando a sintaxe do sistema de arquivos do 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
```

Para obter mais informações sobre o upload de objetos, *Uploading Objects in the Guia do desenvolvedor do Amazon S3*.

- Para API obter detalhes, consulte [PutObject](#) na Referência de AWS CLI Comandos.

put-public-access-block

O código de exemplo a seguir mostra como usar `put-public-access-block`.

AWS CLI

Para definir a configuração do bloco de acesso público para um bucket

O `put-public-access-block` exemplo a seguir define uma configuração restritiva de acesso público de blocos para o bucket especificado.

```
aws s3api put-public-access-block \  
  --bucket my-bucket \  
  --public-access-block-  
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutPublicAccessBlock](#) na Referência de AWS CLI Comandos.

rb

O código de exemplo a seguir mostra como usar `rb`.

AWS CLI

Exemplo 1: excluir um bucket

O `rb` comando a seguir remove um bucket. Neste exemplo, o bucket do usuário é `mybucket`. Observe que o balde deve estar vazio para remover:

```
aws s3 rb s3://mybucket
```

Saída:

```
remove_bucket: mybucket
```

Exemplo 2: forçar a exclusão de um bucket

O `rb` comando a seguir usa o `--force` parâmetro para primeiro remover todos os objetos no bucket e depois remover o próprio bucket. Neste exemplo, o bucket do usuário é `mybucket` e os objetos nele `mybucket` são `test1.txt` e `test2.txt`:

```
aws s3 rb s3://mybucket \  
--force
```

Saída:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt  
remove_bucket: mybucket
```

- Para API obter detalhes, consulte [Rb](#) na Referência de AWS CLI Comandos.

restore-object

O código de exemplo a seguir mostra como usar `restore-object`.

AWS CLI

Como criar uma solicitação de restauração para um objeto

O exemplo de `restore-object` a seguir restaura o objeto Amazon S3 Glacier especificado do bucket `my-glacier-bucket` por 10 dias.

```
aws s3api restore-object \  
--bucket my-glacier-bucket \  
--key my-glacier-object \  
--restore-type CopyAndReplace \  
--days 10
```

```
--bucket my-glacier-bucket \  
--key doc1.rtf \  
--restore-request Days=10
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RestoreObject](#) na Referência de AWS CLI Comandos.

rm

O código de exemplo a seguir mostra como usar `rm`.

AWS CLI

Exemplo 1: Excluir um objeto do S3

O `rm` comando a seguir exclui um único objeto s3:

```
aws s3 rm s3://mybucket/test2.txt
```

Saída:

```
delete: s3://mybucket/test2.txt
```

Exemplo 2: excluir todo o conteúdo em um bucket

O `rm` comando a seguir exclui recursivamente todos os objetos em um bucket e prefixo especificados quando transmitido com o parâmetro. `--recursive` Neste exemplo, o bucket `mybucket` contém os objetos `test1.txt` e `test2.txt`:

```
aws s3 rm s3://mybucket \  
--recursive
```

Saída:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt
```

Exemplo 3: Exclua todo o conteúdo em um bucket, exceto os arquivos ``.jpg``

O `rm` comando a seguir exclui recursivamente todos os objetos em um bucket e prefixo especificados quando transmitidos com o parâmetro, `--recursive` enquanto exclui alguns objetos usando um parâmetro. `--exclude` Neste exemplo, o bucket `mybucket` tem os objetos `test1.txt` e `test2.jpg`:

```
aws s3 rm s3://mybucket/ \  
--recursive \  
--exclude "*.jpg"
```

Saída:

```
delete: s3://mybucket/test1.txt
```

Exemplo 4: excluir todo o conteúdo em um bucket, exceto objetos com o prefixo especificado

O `rm` comando a seguir exclui recursivamente todos os objetos em um bucket e prefixo especificados quando transmitidos com o parâmetro, `--recursive` enquanto exclui todos os objetos em um determinado prefixo usando um parâmetro. `--exclude` Neste exemplo, o bucket `mybucket` tem os objetos `test1.txt` e `another/test.txt`:

```
aws s3 rm s3://mybucket/ \  
--recursive \  
--exclude "another/*"
```

Saída:

```
delete: s3://mybucket/test1.txt
```

Exemplo 5: excluir um objeto de um ponto de acesso do S3

O `rm` comando a seguir exclui um único objeto (`mykey`) do ponto de acesso (`myaccesspoint`). ::
O `rm` comando a seguir exclui um único objeto (`mykey`) do ponto de acesso (`myaccesspoint`).

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

Saída:

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```


- Para API obter detalhes, consulte [Rm](#) na Referência de AWS CLI Comandos.

select-object-content

O código de exemplo a seguir mostra como usar `select-object-content`.

AWS CLI

Para filtrar o conteúdo de um objeto do Amazon S3 com base em uma declaração SQL

O `select-object-content` exemplo a seguir filtra o objeto `my-data-file.csv` com a SQL instrução especificada e envia a saída para um arquivo.

```
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"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SelectObjectContent](#) na Referência de AWS CLI Comandos.

sync

O código de exemplo a seguir mostra como usar `sync`.

AWS CLI

Exemplo 1: sincronizar todos os objetos locais com o bucket especificado

O `sync` comando a seguir sincroniza objetos de um diretório local com o prefixo e o bucket especificados fazendo o upload dos arquivos locais para o S3. Um arquivo local exigirá o upload se o tamanho do arquivo local for diferente do tamanho do objeto do S3, se a hora da última modificação do arquivo local for mais recente que a hora da última modificação do objeto do S3 ou se o arquivo local não existir sob o bucket e o prefixo especificados. Neste exemplo, o usuário sincroniza o bucket `mybucket` com o diretório atual local. O diretório atual local contém os arquivos `test.txt` `test2.txt` e. O bucket `mybucket` contém objetos.

```
aws s3 sync . s3://mybucket
```

Saída:

```
upload: test.txt to s3://mybucket/test.txt
upload: test2.txt to s3://mybucket/test2.txt
```

Exemplo 2: sincronizar todos os objetos do S3 do bucket do S3 especificado com outro bucket

O sync comando a seguir sincroniza objetos sob um prefixo e bucket especificados com objetos sob outro prefixo e bucket especificados copiando objetos do S3. Um objeto do S3 precisará ser copiado se os tamanhos dos dois objetos do S3 forem diferentes, se a hora da última modificação da origem for mais recente que a hora da última modificação do destino ou se o objeto do S3 não existir no destino do bucket e do prefixo especificados.

Neste exemplo, o usuário sincroniza o bucket mybucket com o bucketmybucket2. O bucket mybucket contém os objetos test.txt test2.txt e. O bucket não mybucket2 contém objetos:

```
aws s3 sync s3://mybucket s3://mybucket2
```

Saída:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
copy: s3://mybucket/test2.txt to s3://mybucket2/test2.txt
```

Exemplo 3: sincronizar todos os objetos do S3 do bucket do S3 especificado com o diretório local

O sync comando a seguir sincroniza arquivos do bucket do S3 especificado com o diretório local baixando objetos do S3. Um objeto do S3 exigirá o download se o tamanho do objeto do S3 for diferente do tamanho do arquivo local, se a hora da última modificação do objeto do S3 for mais recente do que a hora da última modificação do arquivo local ou se o objeto do S3 não existir no diretório local. Observe que quando os objetos são baixados do S3, a hora da última modificação do arquivo local é alterada para a hora da última modificação do objeto do S3. Neste exemplo, o usuário sincroniza o bucket mybucket com o diretório local atual. O bucket mybucket contém os objetos test.txt test2.txt e. O diretório local atual não tem arquivos:

```
aws s3 sync s3://mybucket .
```

Saída:

```
download: s3://mybucket/test.txt to test.txt
download: s3://mybucket/test2.txt to test2.txt
```

Exemplo 4: sincronizar todos os objetos locais com o bucket especificado e excluir todos os arquivos que não correspondam

O sync comando a seguir sincroniza objetos sob um prefixo e bucket especificados com arquivos em um diretório local fazendo o upload dos arquivos locais para o S3. Por causa do `--delete` parâmetro, todos os arquivos existentes sob o prefixo e o bucket especificados, mas não existentes no diretório local, serão excluídos. Neste exemplo, o usuário sincroniza o bucket mybucket com o diretório atual local. O diretório atual local contém os arquivos `test.txt` `test2.txt` e. O bucket mybucket contém o objeto `test3.txt`:

```
aws s3 sync . s3://mybucket \
--delete
```

Saída:

```
upload: test.txt to s3://mybucket/test.txt
upload: test2.txt to s3://mybucket/test2.txt
delete: s3://mybucket/test3.txt
```

Exemplo 5: Sincronizar todos os objetos locais com o bucket especificado, exceto os arquivos ``.jpg``

O sync comando a seguir sincroniza objetos sob um prefixo e bucket especificados com arquivos em um diretório local fazendo o upload dos arquivos locais para o S3. Por causa do `--exclude` parâmetro, todos os arquivos que correspondam ao padrão existente no S3 e localmente serão excluídos da sincronização. Neste exemplo, o usuário sincroniza o bucket mybucket com o diretório atual local. O diretório atual local contém os arquivos `test.jpg` `test2.txt` e. O bucket mybucket contém o objeto `test.jpg` de um tamanho diferente do `localtest.jpg`:

```
aws s3 sync . s3://mybucket \
--exclude "*.jpg"
```

Saída:

```
upload: test2.txt to s3://mybucket/test2.txt
```

Exemplo 6: Sincronizar todos os objetos locais com o bucket especificado, exceto os arquivos ``.jpg``

O sync comando a seguir sincroniza arquivos em um diretório local com objetos com um prefixo e bucket especificados baixando objetos do S3. Este exemplo usa o sinalizador de `--exclude` parâmetros para excluir um diretório especificado e o prefixo S3 do sync comando. Neste exemplo, o usuário sincroniza o diretório atual local com o bucket `mybucket`. O diretório atual local contém os arquivos `test.txt` `another/test2.txt` e. O bucket `mybucket` contém os objetos `another/test5.txt` e `test1.txt`:

```
aws s3 sync s3://mybucket/ . \  
--exclude "*another/*"
```

Saída:

```
download: s3://mybucket/test1.txt to test1.txt
```

Exemplo 7: sincronizar todos os objetos entre buckets em diferentes regiões

O sync comando a seguir sincroniza arquivos entre dois buckets em regiões diferentes:

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \  
--source-region us-west-2 \  
--region us-east-1
```

Saída:

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

Exemplo 8: Sincronizar com um ponto de acesso S3

O sync comando a seguir sincroniza o diretório atual com o ponto de acesso (`myaccesspoint`):

```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Saída:

```
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
```

- Para API obter detalhes, consulte [Sincronizar](#) na Referência de AWS CLI Comandos.

upload-part-copy

O código de exemplo a seguir mostra como usar `upload-part-copy`.

AWS CLI

Para carregar parte de um objeto copiando dados de um objeto existente como fonte de dados

O `upload-part-copy` exemplo a seguir carrega uma peça copiando dados de um objeto existente como fonte de dados.

```
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"
```

Saída:

```
{
  "CopyPartResult": {
    "LastModified": "2019-12-13T23:16:03.000Z",
    "ETag": "\"711470fc377698c393d94aed6305e245\""
  }
}
```

- Para API obter detalhes, consulte [UploadPartCopy](#) na Referência de AWS CLI Comandos.

upload-part

O código de exemplo a seguir mostra como usar `upload-part`.

AWS CLI

O seguinte comando faz o upload da primeira parte de um upload fracionado iniciado com o comando `create-multipart-upload`:

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --  
body part01 --upload-id  
"dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR"
```

A opção `body` usa o nome ou o caminho de um arquivo local para upload (não use o prefixo `file://`). O tamanho mínimo de uma parte é de 5 MB. O ID de upload é retornado por `create-multipart-upload` e também pode ser recuperado com `list-multipart-uploads`. O bucket e a chave são especificados ao criar o upload fracionado.

Saída:

```
{  
  "ETag": "\"e868e0f4719e394144ef36531ee6824c\""  
}
```

Guarde o ETag valor de cada peça para mais tarde. Eles são necessários para concluir o upload fracionado.

- Para API obter detalhes, consulte [UploadPart](#) na Referência de AWS CLI Comandos.

website

O código de exemplo a seguir mostra como usar `website`.

AWS CLI

Configurar um bucket do S3 como um site estático

O comando a seguir configura um bucket `my-bucket` chamado site estático. A opção de documento de índice especifica o arquivo para `my-bucket` o qual os visitantes serão direcionados quando navegarem até o siteURL. Nesse caso, o bucket está na região `us-west-2`, então o site apareceria em. `http://my-bucket.s3-website-us-west-2.amazonaws.com`

Todos os arquivos no bucket que aparecem no site estático devem ser configurados para permitir que os visitantes os abram. As permissões de arquivo são configuradas separadamente da configuração do site do bucket.

```
aws s3 website s3://my-bucket/ \  
--index-document index.html \  
--error-document error.html
```

Para obter informações sobre como hospedar um site estático no Amazon S3, consulte [Hospedagem de um site estático no Guia do](#) desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [Website](#) in AWS CLI Command Reference.

Exemplos de controle do Amazon S3 usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o controle do AWS Command Line Interface Amazon S3.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-access-point

O código de exemplo a seguir mostra como usar `create-access-point`.

AWS CLI

Para criar um ponto de acesso

O `create-access-point` exemplo a seguir cria um ponto de acesso com o nome `finance-ap` do bucket `business-records` na conta `123456789012`. Antes de executar esse exemplo, substitua o nome do ponto de acesso, o nome do bucket e o número da conta pelos valores apropriados para seu caso de uso.

```
aws s3control create-access-point \  
--bucket-name business-records \  
--access-point-name finance-ap \  
--account-id 123456789012
```

```
--account-id 123456789012 \  
--bucket business-records \  
--name finance-ap
```

Este comando não produz saída.

Para obter mais informações, consulte [Criação de pontos de acesso](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [CreateAccessPoint](#) na Referência de AWS CLI Comandos.

create-job

O código de exemplo a seguir mostra como usar create-job.

AWS CLI

Para criar um trabalho de operações em lote do Amazon S3

O create-job exemplo a seguir cria um trabalho de operações em lote do Amazon S3 para marcar objetos como `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
```

Saída:

```
{  
  "JobId": "93735294-df46-44d5-8638-6356f335324e"  
}
```


- Para API obter detalhes, consulte [CreateJob](#) na Referência de AWS CLI Comandos.

delete-access-point-policy

O código de exemplo a seguir mostra como usar `delete-access-point-policy`.

AWS CLI

Para excluir uma política de ponto de acesso

O `delete-access-point-policy` exemplo a seguir exclui a política de ponto de acesso do ponto de acesso nomeado `finance-ap` na conta `123456789012`. Antes de executar este exemplo, substitua o nome do ponto de acesso e o número da conta pelos valores apropriados para seu caso de uso.

```
aws s3control delete-access-point-policy \  
  --account-id 123456789012 \  
  --name finance-ap
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do acesso a dados com pontos de acesso do Amazon S3 no Guia](#) do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [DeleteAccessPointPolicy](#) na Referência de AWS CLI Comandos.

delete-access-point

O código de exemplo a seguir mostra como usar `delete-access-point`.

AWS CLI

Para excluir um ponto de acesso

O `delete-access-point` exemplo a seguir exclui um ponto de acesso chamado `finance-ap` na conta `123456789012`. Antes de executar este exemplo, substitua o nome do ponto de acesso e o número da conta pelos valores apropriados para seu caso de uso.

```
aws s3control delete-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

```
--name finance-ap
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do acesso a dados com pontos de acesso do Amazon S3 no Guia](#) do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [DeleteAccessPoint](#) na Referência de AWS CLI Comandos.

delete-public-access-block

O código de exemplo a seguir mostra como usar `delete-public-access-block`.

AWS CLI

Para excluir as configurações de bloqueio de acesso público de uma conta

O `delete-public-access-block` exemplo a seguir exclui as configurações de bloqueio de acesso público para a conta especificada.

```
aws s3control delete-public-access-block \  
  --account-id 123456789012
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePublicAccessBlock](#) na Referência de AWS CLI Comandos.

describe-job

O código de exemplo a seguir mostra como usar `describe-job`.

AWS CLI

Para descrever um trabalho de operações em lote do Amazon S3

O seguinte `describe-job` fornece parâmetros de configuração e status para o trabalho de operações em lote especificado.

```
aws s3control describe-job \  
  --account-id 123456789012 \  
  --job-id 93735294-df46-44d5-8638-6356f335324e
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [DescribeJob](#) na Referência de AWS CLI Comandos.

get-access-point-policy-status

O código de exemplo a seguir mostra como usar `get-access-point-policy-status`.

AWS CLI

Para recuperar o status da política do ponto de acesso

O `get-access-point-policy-status` exemplo a seguir recupera o status da política de ponto de acesso para o ponto de acesso nomeado `finance-ap` na conta `123456789012`. O status da política do ponto de acesso indica se a política do ponto de acesso permite acesso público. Antes de executar este exemplo, substitua o nome do ponto de acesso e o número da conta pelos valores apropriados para seu caso de uso.

```
aws s3control get-access-point-policy-status \
  --account-id 123456789012 \
  --name finance-ap
```

Saída:

```
{
  "PolicyStatus": {
    "IsPublic": false
  }
}
```

Para obter mais informações sobre quando uma política de ponto de acesso é considerada pública, consulte [The Meaning of “Public”](#) no Amazon Simple Storage Service Developer Guide.

- Para API obter detalhes, consulte [GetAccessPointPolicyStatus](#) na Referência de AWS CLI Comandos.

get-access-point-policy

O código de exemplo a seguir mostra como usar `get-access-point-policy`.

AWS CLI

Para recuperar uma política de ponto de acesso

O `get-access-point-policy` exemplo a seguir recupera a política de ponto de acesso do ponto de acesso nomeado `finance-ap` na conta `123456789012`. Antes de executar este exemplo, substitua o nome do ponto de acesso e o número da conta pelos valores apropriados para seu caso de uso.

```
aws s3control get-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap
```

Saída:

```
{
  "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/*\"}]}"
}
```

Para obter mais informações, consulte [Gerenciamento do acesso a dados com pontos de acesso do Amazon S3 no Guia](#) do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [GetAccessPointPolicy](#) na Referência de AWS CLI Comandos.

get-access-point

O código de exemplo a seguir mostra como usar `get-access-point`.

AWS CLI

Para recuperar detalhes da configuração do ponto de acesso

O `get-access-point` exemplo a seguir recupera os detalhes da configuração do ponto de acesso nomeado `finance-ap` na conta `123456789012`. Antes de executar este exemplo,

substitua o nome do ponto de acesso e o número da conta pelos valores apropriados para seu caso de uso.

```
aws s3control get-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

Saída:

```
{  
  "Name": "finance-ap",  
  "Bucket": "business-records",  
  "NetworkOrigin": "Internet",  
  "PublicAccessBlockConfiguration": {  
    "BlockPublicAcls": false,  
    "IgnorePublicAcls": false,  
    "BlockPublicPolicy": false,  
    "RestrictPublicBuckets": false  
  },  
  "CreationDate": "2020-01-01T00:00:00Z"  
}
```

Para obter mais informações, consulte [Gerenciamento do acesso a dados com pontos de acesso do Amazon S3 no Guia](#) do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [GetAccessPoint](#) na Referência de AWS CLI Comandos.

get-multi-region-access-point-routes

O código de exemplo a seguir mostra como usar `get-multi-region-access-point-routes`.

AWS CLI

Para consultar a configuração atual da rota do ponto de acesso multirregional

O `get-multi-region-access-point-routes` exemplo a seguir retorna a configuração de roteamento atual para o ponto de acesso multirregional especificado.

```
aws s3control get-multi-region-access-point-routes \  
  --region Region \  
  --account-id 111122223333 \  
  --access-point ap-123456789012
```

```
--mrap MultiRegionAccessPoint_ARN
```

Saída:

```
{
  "Mrap": "arn:aws:s3::111122223333:accesspoint/0000000000000000.mrap",
  "Routes": [
    {
      "Bucket": "DOC-EXAMPLE-BUCKET-1",
      "Region": "ap-southeast-2",
      "TrafficDialPercentage": 100
    },
    {
      "Bucket": "DOC-EXAMPLE-BUCKET-2",
      "Region": "us-west-1",
      "TrafficDialPercentage": 0
    }
  ]
}
```

- Para API obter detalhes, consulte [GetMultiRegionAccessPointRoutes](#) na Referência de AWS CLI Comandos.

get-public-access-block

O código de exemplo a seguir mostra como usar `get-public-access-block`.

AWS CLI

Para listar as configurações públicas de bloqueio de acesso de uma conta

O `get-public-access-block` exemplo a seguir exibe as configurações de bloqueio de acesso público para a conta especificada.

```
aws s3control get-public-access-block \
  --account-id 123456789012
```

Saída:

```
{
```

```
"PublicAccessBlockConfiguration": {
  "BlockPublicPolicy": true,
  "RestrictPublicBuckets": true,
  "IgnorePublicAcls": true,
  "BlockPublicAcls": true
}
```

- Para API obter detalhes, consulte [GetPublicAccessBlock](#) na Referência de AWS CLI Comandos.

list-access-points

O código de exemplo a seguir mostra como usar `list-access-points`.

AWS CLI

Exemplo 1: Para recuperar uma lista de todos os pontos de acesso de uma conta

O `list-access-points` exemplo a seguir exibe uma lista de todos os pontos de acesso conectados aos buckets pertencentes à conta 123456789012.

```
aws s3control list-access-points \
  --account-id 123456789012
```

Saída:

```
{
  "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"
  }
]
```

Exemplo 2: Para recuperar uma lista de todos os pontos de acesso de um bucket

O `list-access-points` exemplo a seguir recupera uma lista de todos os pontos de acesso anexados ao bucket de `external-docs` propriedade da conta `123456789012`.

```
aws s3control list-access-points \
  --account-id 123456789012 \
  --bucket external-docs
```

Saída:

```
{
  "AccessPointList": [
    {
      "Name": "customer-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    },
    {
      "Name": "public-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    }
  ]
}
```

```
]
}
```

Para obter mais informações, consulte [Gerenciamento do acesso a dados com pontos de acesso do Amazon S3 no Guia](#) do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [ListAccessPoints](#) na Referência de AWS CLI Comandos.

list-jobs

O código de exemplo a seguir mostra como usar `list-jobs`.

AWS CLI

Para listar os trabalhos de operações em lote do Amazon S3 de uma conta

O `list-jobs` exemplo a seguir lista todos os trabalhos recentes de operações em lote para a conta especificada.

```
aws s3control list-jobs \
  --account-id 123456789012
```

Saída:

```
{
  "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
  },
]
}

```

- Para API obter detalhes, consulte [ListJobs](#) na Referência de AWS CLI Comandos.

put-access-point-policy

O código de exemplo a seguir mostra como usar `put-access-point-policy`.

AWS CLI

Para definir uma política de ponto de acesso

O `put-access-point-policy` exemplo a seguir coloca a política de ponto de acesso especificada para o ponto de acesso `finance-ap` na conta `123456789012`. Se o ponto de acesso `finance-ap` já tiver uma política, esse comando substituirá a política existente pela especificada nesse comando. Antes de executar este exemplo, substitua o número da conta, o nome do ponto de acesso e as declarações de política pelos valores apropriados para seu caso de uso.

```

aws s3control put-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap \
  --policy file://ap-policy.json

```

Conteúdo de `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/*"
    }
]
}

```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento do acesso a dados com pontos de acesso do Amazon S3 no Guia](#) do desenvolvedor do Amazon Simple Storage Service.

- Para API obter detalhes, consulte [PutAccessPointPolicy](#) na Referência de AWS CLI Comandos.

put-public-access-block

O código de exemplo a seguir mostra como usar `put-public-access-block`.

AWS CLI

Para editar as configurações de bloqueio de acesso público de uma conta

O `put-public-access-block` exemplo a seguir alterna todas as configurações de bloqueio de acesso público `true` para a conta especificada.

```

aws s3control put-public-access-block \
  --account-id 123456789012 \
  --public-access-block-configuration '{"BlockPublicAcls": true,
"IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
true}'

```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutPublicAccessBlock](#) na Referência de AWS CLI Comandos.

submit-multi-region-access-point-routes

O código de exemplo a seguir mostra como usar `submit-multi-region-access-point-routes`.

AWS CLI

Para atualizar sua configuração de roteamento de pontos de acesso multirregionais

O `submit-multi-region-access-point-routes` exemplo a seguir atualiza os status de roteamento de `DOC-EXAMPLE-BUCKET-1` e `DOC-EXAMPLE-BUCKET-2` na `ap-southeast-2` região do seu ponto de acesso multirregional.

```
aws s3control submit-multi-region-access-point-routes \  
  --region ap-southeast-2 \  
  --account-id 111122223333 \  
  --mrap MultiRegionAccessPoint_ARN \  
  --route-updates Bucket=DOC-EXAMPLE-  
BUCKET-1,TrafficDialPercentage=100 Bucket=DOC-EXAMPLE-  
BUCKET-2,TrafficDialPercentage=0
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SubmitMultiRegionAccessPointRoutes](#) na Referência de AWS CLI Comandos.

update-job-priority

O código de exemplo a seguir mostra como usar `update-job-priority`.

AWS CLI

Para atualizar a prioridade de um trabalho de operações em lote do Amazon S3

O `update-job-priority` exemplo a seguir atualiza o trabalho especificado para uma nova prioridade.

```
aws s3control update-job-priority \  
  --account-id 123456789012 \  
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
  --priority 52
```

Saída:

```
{  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",
```

```
"Priority": 52
}
```

- Para API obter detalhes, consulte [UpdateJobPriority](#) na Referência de AWS CLI Comandos.

update-job-status

O código de exemplo a seguir mostra como usar `update-job-status`.

AWS CLI

Para atualizar o status de um trabalho de operações em lote do Amazon S3

O `update-job-status` exemplo a seguir cancela o trabalho especificado que está aguardando aprovação.

```
aws s3control update-job-status \
  --account-id 123456789012 \
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
  --requested-job-status Cancelled
```

Saída:

```
{
  "Status": "Cancelled",
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"
}
```

O `update-job-status` exemplo a seguir confirma e executa o especificado que está aguardando aprovação.

```
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"  
}
```

O `update-job-status` exemplo a seguir cancela o trabalho especificado que está sendo executado.

```
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"  
}
```

- Para API obter detalhes, consulte [UpdateJobStatus](#) na Referência de AWS CLI Comandos.

Exemplos do S3 Glacier usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o S3 Glacier.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

abort-multipart-upload

O código de exemplo a seguir mostra como usar `abort-multipart-upload`.

AWS CLI

O comando a seguir exclui um upload de várias partes em andamento para um cofre chamado: `my-vault`

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
--upload-id 19gaRezEXAMPLES6Ry5YYdqthH0C_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ
```

Esse comando não produz nenhuma saída. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso. O ID de upload é retornado pelo comando `aws glacier initiate-multipart-upload` e também pode ser coletado usando `aws glacier list-multipart-uploads`.

Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando AWS CLI o, consulte [Como usar o Amazon Glacier no Guia do usuário.AWS CLI](#)

- Para API obter detalhes, consulte [AbortMultipartUpload](#) na Referência de AWS CLI Comandos.

abort-vault-lock

O código de exemplo a seguir mostra como usar `abort-vault-lock`.

AWS CLI

Para abortar um processo de bloqueio de cofre em andamento

O `abort-vault-lock` exemplo a seguir exclui uma política de bloqueio de cofre do cofre especificado e redefine o estado de bloqueio do cofre para desbloqueado.

```
aws glacier abort-vault-lock \
--account-id - \
--vault-name MyVaultName
```

Este comando não produz saída.

Para obter mais informações, consulte [Abort Vault DELETE Lock \(política de bloqueio\) no Guia do desenvolvedor do Amazon Glacier. API](#)

- Para API obter detalhes, consulte [AbortVaultLock](#) na Referência de AWS CLI Comandos.

add-tags-to-vault

O código de exemplo a seguir mostra como usar `add-tags-to-vault`.

AWS CLI

O seguinte comando adiciona duas tags ao cofre `my-vault`:

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --  
tags id=1234,date=july2015
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [AddTagsToVault](#) na Referência de AWS CLI Comandos.

complete-multipart-upload

O código de exemplo a seguir mostra como usar `complete-multipart-upload`.

AWS CLI

O comando a seguir conclui o upload de várias partes para um arquivo de 3 MiB:

```
aws glacier complete-multipart-upload --archive-size 3145728 --  
checksum 9628195fcdcbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67  
--upload-id 19gaRezEXAMPLES6Ry5YYdqthH0C_kGRCT03L9yetr220UmPtBYKk-  
0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

O ID de upload é retornado pelo comando `aws glacier initiate-multipart-upload` e também pode ser coletado usando `aws glacier list-multipart-uploads`. O parâmetro `checksum` usa um hash de árvore SHA -256 do arquivo em hexadecimal.

Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando AWS CLI o, incluindo instruções sobre o cálculo de um hash de árvore, consulte [Como usar o Amazon Glacier no Guia do usuário](#). AWS CLI

- Para API obter detalhes, consulte [CompleteMultipartUpload](#) na Referência de AWS CLI Comandos.

complete-vault-lock

O código de exemplo a seguir mostra como usar `complete-vault-lock`.

AWS CLI

Para concluir um processo de bloqueio de cofre em andamento

O `complete-vault-lock` exemplo a seguir completa o andamento do bloqueio em andamento para o cofre especificado e define o estado de bloqueio do cofre como `Locked`. Você obtém o valor do `lock-id` parâmetro ao executar `initiate-lock-process`.

```
aws glacier complete-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName \  
  --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Complete Vault Lock \(POSTlockId\) no Guia](#) do desenvolvedor do Amazon API Glacier.

- Para API obter detalhes, consulte [CompleteVaultLock](#) na Referência de AWS CLI Comandos.

create-vault

O código de exemplo a seguir mostra como usar `create-vault`.

AWS CLI

O seguinte comando cria um cofre chamado `my-vault`:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [CreateVault](#) na Referência de AWS CLI Comandos.

delete-archive

O código de exemplo a seguir mostra como usar `delete-archive`.

AWS CLI

Como excluir um arquivo de um cofre

O exemplo de `delete-archive` a seguir remove o arquivamento especificado de `example_vault`.

```
aws glacier delete-archive \  
  --account-id 111122223333 \  
  --vault-name example_vault \  
  --archive-id Sc0u9ZP8yaWkmh-XGLIvAVprtLhaLCGnNwNL5I5x9HqPIkX5mjc0DrId3Ln-  
  Gi_k2HzmLIDZUz117KSdVMdMXLuFWi9PJUitxW073edQ43eTLMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteArchive](#) na Referência de AWS CLI Comandos.

`delete-vault-access-policy`

O código de exemplo a seguir mostra como usar `delete-vault-access-policy`.

AWS CLI

Para remover a política de acesso de um cofre

O `delete-vault-access-policy` exemplo a seguir remove a política de acesso do cofre especificado.

```
aws glacier delete-vault-access-policy \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteVaultAccessPolicy](#) na Referência de AWS CLI Comandos.

`delete-vault-notifications`

O código de exemplo a seguir mostra como usar `delete-vault-notifications`.

AWS CLI

Para remover as SNS notificações de um cofre

O `delete-vault-notifications` exemplo a seguir remove as notificações enviadas pelo Amazon Simple Notification Service (AmazonSNS) para o cofre especificado.

```
aws glacier delete-vault-notifications \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteVaultNotifications](#) na Referência de AWS CLI Comandos.

delete-vault

O código de exemplo a seguir mostra como usar `delete-vault`.

AWS CLI

O seguinte comando cria um cofre chamado `my-vault`:

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

Esse comando não produz nenhuma saída. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [DeleteVault](#) na Referência de AWS CLI Comandos.

describe-job

O código de exemplo a seguir mostra como usar `describe-job`.

AWS CLI

O seguinte comando recupera informações sobre um trabalho de recuperação de inventário no cofre `my-vault`:

```
aws glacier describe-job --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqLNHS61ds04CnMW
```

Saída:

```
{
  "InventoryRetrievalParameters": {
    "Format": "JSON"
  },
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "Completed": false,
  "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqLNHS61ds04CnMW",
  "Action": "InventoryRetrieval",
  "CreationDate": "2015-07-17T20:23:41.616Z",
  "StatusCode": "InProgress"
}
```

O ID do trabalho pode ser encontrado na saída de `aws glacier initiate-job` e `aws glacier list-jobs`. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [DescribeJob](#) na Referência de AWS CLI Comandos.

describe-vault

O código de exemplo a seguir mostra como usar `describe-vault`.

AWS CLI

O seguinte comando recupera dados sobre o cofre `my-vault`:

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [DescribeVault](#) na Referência de AWS CLI Comandos.

get-data-retrieval-policy

O código de exemplo a seguir mostra como usar `get-data-retrieval-policy`.

AWS CLI

O comando a seguir obtém a política de recuperação de dados para a conta em uso:

```
aws glacier get-data-retrieval-policy --account-id -
```

Saída:

```
{
  "Policy": {
    "Rules": [
      {
        "BytesPerHour": 10737418240,
        "Strategy": "BytesPerHour"
      }
    ]
  }
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [GetDataRetrievalPolicy](#) na Referência de AWS CLI Comandos.

get-job-output

O código de exemplo a seguir mostra como usar `get-job-output`.

AWS CLI

O seguinte comando salva a saída de um trabalho de inventário do cofre em um arquivo no diretório atual chamado `output.json`:

```
aws glacier get-job-output --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdBgf_Q2DK5Ejh18CnTS5XW4_XqLNHS61ds04CnMW output.json
```

O `job-id` está disponível na saída de `aws glacier list-jobs`. Observe que o nome do arquivo de saída é um argumento posicional que não é prefixado por um nome de opção. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Saída:

```
{
  "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_mw6k76ZFGFIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUMZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw","ArchiveDescription":"multipart
upload
test","CreationDate":"2015-04-06T22:24:34Z","Size":3145728,"SHA256TreeHash":"9628195fcdcbcb
```

- Para API obter detalhes, consulte [GetJobOutput](#) na Referência de AWS CLI Comandos.

get-vault-access-policy

O código de exemplo a seguir mostra como usar `get-vault-access-policy`.

AWS CLI

Para recuperar a política de acesso de um cofre

O `get-vault-access-policy` exemplo a seguir recupera a política de acesso para o cofre especificado.

```
aws glacier get-vault-access-policy \
  --account-id 111122223333 \
  --vault-name example_vault
```

Saída:

```
{
  "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\"}]}"
  }
}
```

- Para API obter detalhes, consulte [GetVaultAccessPolicy](#) na Referência de AWS CLI Comandos.

get-vault-lock

O código de exemplo a seguir mostra como usar `get-vault-lock`.

AWS CLI

Para obter os detalhes de uma trava de cofre

O `get-vault-lock` exemplo a seguir recuperou os detalhes sobre a trava do cofre especificado.

```
aws glacier get-vault-lock \
  --account-id - \
  --vault-name MyVaultName
```

Saída:

```
{
  "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"
}
```


Para obter mais informações, consulte [Get Vault GET Lock \(política de bloqueio\)](#) no Guia do desenvolvedor do Amazon Glacier. API

- Para API obter detalhes, consulte [GetVaultLock](#) na Referência de AWS CLI Comandos.

get-vault-notifications

O código de exemplo a seguir mostra como usar `get-vault-notifications`.

AWS CLI

O seguinte comando captura uma descrição da configuração de notificação do cofre `my-vault`:

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

Saída:

```
{
  "vaultNotificationConfig": {
    "Events": [
      "InventoryRetrievalCompleted",
      "ArchiveRetrievalCompleted"
    ],
    "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
  }
}
```

Se nenhuma notificação tiver sido configurada para o cofre, será retornado um erro. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [GetVaultNotifications](#) na Referência de AWS CLI Comandos.

initiate-job

O código de exemplo a seguir mostra como usar `initiate-job`.

AWS CLI

O comando a seguir inicia um trabalho para obter um inventário do cofre: `my-vault`

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
'{"Type": "inventory-retrieval"}'
```

Saída:

```
{
  "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
  "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW"
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

O comando a seguir inicia um trabalho para recuperar um arquivo do cofre: `my-vault`

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

`job-archive-retrieval.json` é um JSON arquivo na pasta local que especifica o tipo de tarefa, ID de arquivamento e alguns parâmetros opcionais:

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-
ybtRDvc2VkJPSDtfKmqRj0IRQLSGsNuDp-
AJV1u2ccmDSyDUmZwKwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
  "Description": "Retrieve archive on 2015-07-17",
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

O arquivo IDs está disponível na saída de `aws glacier upload-archive` `aws glacier get-job-output` e.

Saída:

```
{
```

```
"location": "/011685312445/vaults/mwunderl/jobs/17IL5-
EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
  "jobId": "17IL5-EkXy205uLYaFdAY0iEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav"
}
```

Consulte [Initiate Job in the Amazon API Glacier Reference](#) para obter detalhes sobre o formato dos parâmetros do trabalho.

- Para API obter detalhes, consulte [InitiateJob](#) na Referência de AWS CLI Comandos.

initiate-multipart-upload

O código de exemplo a seguir mostra como usar `initiate-multipart-upload`.

AWS CLI

O comando a seguir inicia um upload de várias partes em um cofre chamado `my-vault` com um tamanho de peça de 1 MiB (1024 x 1024 bytes) por arquivo:

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-
name my-vault --archive-description "multipart upload test"
```

O parâmetro de descrição do arquivo é opcional. O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Esse comando gera uma ID de upload quando bem-sucedido. Use o ID de upload ao fazer o upload de cada parte do seu arquivo com `aws glacier upload-multipart-part`. Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando AWS CLI o, consulte [Como usar o Amazon Glacier no Guia do usuário.AWS CLI](#)

- Para API obter detalhes, consulte [InitiateMultipartUpload](#) na Referência de AWS CLI Comandos.

initiate-vault-lock

O código de exemplo a seguir mostra como usar `initiate-vault-lock`.

AWS CLI

Para iniciar o processo de bloqueio do cofre

O `initiate-vault-lock` exemplo a seguir instala uma política de bloqueio de cofre no cofre especificado e define o estado de bloqueio do cofre como `InProgress`. Você deve concluir o processo ligando `complete-vault-lock` dentro de 24 horas para definir o estado do bloqueio do cofre como `Locked`.

```
aws glacier initiate-vault-lock \
  --account-id - \
  --vault-name MyVaultName \
  --policy file://vault_lock_policy.json
```

Conteúdo de `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\"}}}]}"}
```

A saída é a ID de bloqueio do cofre que você pode usar para concluir o processo de bloqueio do cofre.

```
{
  "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"
}
```

Para obter mais informações, consulte [Initiate Vault POST Lock \(política de bloqueio\) no Guia do desenvolvedor do Amazon Glacier](#). API

- Para API obter detalhes, consulte [InitiateVaultLock](#) na Referência de AWS CLI Comandos.

list-jobs

O código de exemplo a seguir mostra como usar `list-jobs`.

AWS CLI

O seguinte comando lista trabalhos em andamento e concluídos recentemente no cofre `my-vault`:

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

Saída:

```
{
  "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":
"9628195fcdbcbbe76cde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
      "JobId": "l7IL5-EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
      "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkJPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
      "JobDescription": "Retrieve archive on 2015-07-17",
      "ArchiveSizeInBytes": 3145728,
      "Action": "ArchiveRetrieval",
      "ArchiveSHA256TreeHash":
"9628195fcdbcbbe76cde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
      "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_Xq1NHS61ds04CnMW",
      "Action": "InventoryRetrieval",
      "CreationDate": "2015-07-17T20:23:41.616Z",
      "StatusCode": ""InProgress""
    }
  ]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [ListJobs](#) na Referência de AWS CLI Comandos.

list-multipart-uploads

O código de exemplo a seguir mostra como usar `list-multipart-uploads`.

AWS CLI

O comando a seguir mostra todos os uploads de várias partes em andamento para um cofre chamado: `my-vault`

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando AWS CLI o, consulte Como usar o Amazon Glacier no Guia do usuário.AWS CLI

- Para API obter detalhes, consulte [ListMultipartUploads](#) na Referência de AWS CLI Comandos.

list-parts

O código de exemplo a seguir mostra como usar `list-parts`.

AWS CLI

O comando a seguir lista as partes enviadas para um upload de várias partes em um cofre chamado: `my-vault`

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdıCs8ut-D"
```

Saída:

```
{
  "MultipartUploadId": "SYZi7qnL-
YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdıCs8ut-
D",
  "Parts": [
    {
      "RangeInBytes": "0-1048575",
      "SHA256TreeHash":
      "e1f2a7cd6e047350f69b9f8cfa60fa606fe2f02802097a9a026360a7edc1f553"
```

```

    },
    {
      "RangeInBytes": "1048576-2097151",
      "SHA256TreeHash":
"43cf3061fb95796aed99a11a6aa3cd8f839eed15e655ab0a597126210636aee6"
    }
  ],
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "CreationDate": "2015-07-18T00:05:23.830Z",
  "PartSizeInBytes": 1048576
}

```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando AWS CLI o, consulte Como usar o Amazon Glacier no Guia do usuário.AWS CLI

- Para API obter detalhes, consulte [ListParts](#) na Referência de AWS CLI Comandos.

list-provisioned-capacity

O código de exemplo a seguir mostra como usar `list-provisioned-capacity`.

AWS CLI

Para recuperar as unidades de capacidade provisionadas

O `list-provisioned-capacity` exemplo a seguir recupera detalhes de qualquer unidade de capacidade provisionada para a conta especificada.

```

aws glacier list-provisioned-capacity \
  --account-id 111122223333

```

Saída:

```

{
  "ProvisionedCapacityList": [
    {
      "CapacityId": "HpASAvfRFiVDb0jMfEIcr8K",
      "ExpirationDate": "2020-03-18T19:59:24.000Z",
      "StartDate": "2020-02-18T19:59:24.912Z"
    }
  ]
}

```

```
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListProvisionedCapacity](#) na Referência de AWS CLI Comandos.

list-tags-for-vault

O código de exemplo a seguir mostra como usar `list-tags-for-vault`.

AWS CLI

O seguinte comando lista as tags aplicadas ao cofre `my-vault`:

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

Saída:

```
{  
  "Tags": {  
    "date": "july2015",  
    "id": "1234"  
  }  
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [ListTagsForVault](#) na Referência de AWS CLI Comandos.

list-vaults

O código de exemplo a seguir mostra como usar `list-vaults`.

AWS CLI

O seguinte comando lista os cofres na conta e na região padrão:

```
aws glacier list-vaults --account-id -
```


Saída:

```
{
  "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"
    }
  ]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [ListVaults](#) na Referência de AWS CLI Comandos.

purchase-provisioned-capacity

O código de exemplo a seguir mostra como usar `purchase-provisioned-capacity`.

AWS CLI

Para comprar uma unidade de capacidade provisionada

O `purchase-provisioned-capacity` exemplo a seguir compra uma unidade de capacidade provisionada.

```
aws glacier purchase-provisioned-capacity \
  --account-id 111122223333
```

Saída:

```
{
  "capacityId": "HpASAUvfRFiVDb0jMfEIcr8K"
}
```

- Para API obter detalhes, consulte [PurchaseProvisionedCapacity](#) na Referência de AWS CLI Comandos.

remove-tags-from-vault

O código de exemplo a seguir mostra como usar `remove-tags-from-vault`.

AWS CLI

O comando a seguir remove uma tag com a chave `date` de um cofre chamado `my-vault`:

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-keys date
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [RemoveTagsFromVault](#) na Referência de AWS CLI Comandos.

set-data-retrieval-policy

O código de exemplo a seguir mostra como usar `set-data-retrieval-policy`.

AWS CLI

O comando a seguir configura uma política de recuperação de dados para a conta em uso:

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-policy.json
```

`data-retrieval-policy.json` é um JSON arquivo na pasta atual que especifica uma política de recuperação de dados:

```
{
  "Rules": [
    {
      "Strategy": "BytesPerHour",
      "BytesPerHour": 10737418240
    }
  ]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

O comando a seguir define a política de recuperação de dados para FreeTier uso em JSON linha:

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules": [{"Strategy": "FreeTier"}]}'
```

Consulte Definir política de recuperação de dados na referência do Amazon API Glacier para obter detalhes sobre o formato da política.

- Para API obter detalhes, consulte [SetDataRetrievalPolicy](#) na Referência de AWS CLI Comandos.

set-vault-access-policy

O código de exemplo a seguir mostra como usar set-vault-access-policy.

AWS CLI

Para definir a política de acesso de um cofre

O set-vault-access-policy exemplo a seguir anexa uma política de permissão ao cofre especificado.

```
aws glacier set-vault-access-policy \
  --account-id 111122223333 \
  --vault-name example_vault
  --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\"}]}\"}'
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetVaultAccessPolicy](#) na Referência de AWS CLI Comandos.

set-vault-notifications

O código de exemplo a seguir mostra como usar set-vault-notifications.

AWS CLI

O comando a seguir configura SNS notificações para um cofre chamado: `my-vault`

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-notification-config file://notificationconfig.json
```

`notificationconfig.json` é um JSON arquivo na pasta atual que especifica um SNS tópico e os eventos a serem publicados:

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

- Para API obter detalhes, consulte [SetVaultNotifications](#) na Referência de AWS CLI Comandos.

upload-archive

O código de exemplo a seguir mostra como usar `upload-archive`.

AWS CLI

O seguinte comando faz upload um arquivo na pasta atual com o nome `archive.zip` para um cofre chamado `my-vault`:

```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

Saída:

```
{
  "archiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-ybtRDvc2VkJPSDtfKmqRj0IRQLSGsNuDp-AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
  "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
  "location": "/0123456789012/vaults/my-vault/archives/kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-ybtRDvc2VkJPSDtfKmqRj0IRQLSGsNuDp-AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw"
```

```
}
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

Para recuperar um arquivamento carregado, inicie um trabalho de recuperação com o comando `aws glacier initiate-job`.

- Para API obter detalhes, consulte [UploadArchive](#) na Referência de AWS CLI Comandos.

upload-multipart-part

O código de exemplo a seguir mostra como usar `upload-multipart-part`.

AWS CLI

O comando a seguir carrega a primeira parte de 1 MiB (1024 x 1024 bytes) de um arquivamento:

```
aws glacier upload-multipart-part --body part1 --range 'bytes  
0-1048575/*' --account-id - --vault-name my-vault --upload-  
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-  
0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ
```

O Amazon Glacier exige um argumento de ID de conta ao realizar operações, mas você pode usar um hífen para especificar a conta em uso.

O parâmetro de corpo segue um caminho para um arquivo da parte no sistema de arquivos local. O parâmetro `range` usa um intervalo de HTTP conteúdo indicando os bytes que a peça ocupa no arquivamento concluído. O ID de upload é retornado pelo comando `aws glacier initiate-multipart-upload` e também pode ser coletado usando `aws glacier list-multipart-uploads`.

Para obter mais informações sobre uploads de várias partes para o Amazon Glacier usando AWS CLI o, consulte [Como usar o Amazon Glacier no Guia do usuário.AWS CLI](#)

- Para API obter detalhes, consulte [UploadMultipartPart](#) na Referência de AWS CLI Comandos.

Exemplos de Secrets Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Secrets Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-get-secret-value

O código de exemplo a seguir mostra como usar `batch-get-secret-value`.

AWS CLI

Exemplo 1: Para recuperar o valor secreto de um grupo de segredos listados por nome

O `batch-get-secret-value` exemplo a seguir obtém os segredos do valor secreto para três segredos.

```
aws secretsmanager batch-get-secret-value \  
  --secret-id-list MySecret1 MySecret2 MySecret3
```

Saída:

```
{  
  "SecretValues": [  
    {  
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-  
a1b2c3",  
      "Name": "MySecret1",  
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",  
      "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-  
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-  
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":  
\"secretsmanagertutorial\"}",  
      "VersionStages": [  
        "AWSCURRENT"      ]  
    }  
  ]  
}
```

```

    ],
    "CreateDate": "1523477145.729"
  },
  {
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
    "Name": "MySecret2",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
    "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-EXAMPLEccccc",
    "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\""}",
    "VersionStages": [
      "AWSCURRENT"
    ],
    "CreateDate": "1373477721.124"
  }
],
"Errors": []
}

```

Para obter mais informações, consulte [Recuperar um grupo de segredos em um lote](#) no Guia do Usuário do AWS Secrets Manager.

Exemplo 2: Para recuperar o valor secreto de um grupo de segredos selecionados pelo filtro

O `batch-get-secret-value` exemplo a seguir obtém os segredos de valor secreto em sua conta que estão MySecret no nome. A filtragem por nome diferencia maiúsculas de minúsculas.

```

aws secretsmanager batch-get-secret-value \
  --filters Key="name",Values="MySecret"

```

Saída:

```

{
  "SecretValues": [
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
      "Name": "MySecret1",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
      "SecretString": "{\"username\":\"diego_ramirez\",\"password\":\"EXAMPLE-
PASSWORD\",\"engine\":\"mysql\",\"host\":\"secretsmanagertutorial.cluster.us-
west-2.rds.amazonaws.com\",\"port\":3306,\"dbClusterIdentifier\":
\"secretsmanagertutorial\"}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1523477145.729"
    },
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
      "Name": "MySecret2",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbbb",
      "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\""}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1673477781.275"
    },
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret3-
a1b2c3",
      "Name": "MySecret3",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEccccc",
      "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\""}",
      "VersionStages": [
        "AWSCURRENT"
      ],
      "CreateDate": "1373477721.124"
    }
  ],
  "Errors": []
}

```


Para obter mais informações, consulte [Recuperar um grupo de segredos em um lote](#) no Guia do Usuário do AWS Secrets Manager.

- Para API obter detalhes, consulte [BatchGetSecretValue](#) na Referência de AWS CLI Comandos.

cancel-rotate-secret

O código de exemplo a seguir mostra como usar `cancel-rotate-secret`.

AWS CLI

Para desativar a rotação automática de um segredo

O `cancel-rotate-secret` exemplo a seguir desativa a rotação automática de um segredo. Para retomar a rotação, ligue `rotate-secret`.

```
aws secretsmanager cancel-rotate-secret \  
  --secret-id MyTestSecret
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Para obter mais informações, consulte [Rotate a secret](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [CancelRotateSecret](#) na Referência de AWS CLI Comandos.

create-secret

O código de exemplo a seguir mostra como usar `create-secret`.

AWS CLI

Exemplo 1: como criar um segredo

O seguinte exemplo de `create-secret` cria um segredo com dois pares de chave/valor.

```
aws secretsmanager create-secret \  
  --name MyTestSecret
```

```
--name MyTestSecret \  
--description "My test secret created with the CLI." \  
--secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"  
}
```

Para obter mais informações, consulte [Criar um segredo](#) no Guia do usuário do Secrets Manager.

Exemplo 2: Para criar um segredo a partir das credenciais em um arquivo JSON

O exemplo de `create-secret` a seguir cria um segredo com base em credenciais em um arquivo. Para obter mais informações, consulte [Carregando AWS CLI parâmetros de um arquivo](#) no Guia AWS CLI do usuário.

```
aws secretsmanager create-secret \  
  --name MyTestSecret \  
  --secret-string file://mycreds.json
```

Conteúdo de `mycreds.json`:

```
{  
  "engine": "mysql",  
  "username": "saanvis",  
  "password": "EXAMPLE-PASSWORD",  
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",  
  "dbname": "myDatabase",  
  "port": "3306"  
}
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",
```

```
"VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Para obter mais informações, consulte [Criar um segredo](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [CreateSecret](#) na Referência de AWS CLI Comandos.

delete-resource-policy

O código de exemplo a seguir mostra como usar `delete-resource-policy`.

AWS CLI

Para excluir a política baseada em recursos anexada a um segredo

O exemplo de `delete-resource-policy` a seguir exclui a política baseada em recurso anexada a um segredo.

```
aws secretsmanager delete-resource-policy \  
  --secret-id MyTestSecret
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Para obter mais informações, consulte [Autenticação e controle de acesso](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [DeleteResourcePolicy](#) na Referência de AWS CLI Comandos.

delete-secret

O código de exemplo a seguir mostra como usar `delete-secret`.

AWS CLI

Exemplo 1: como excluir um segredo

O exemplo de `delete-secret` a seguir exclui um segredo. É possível recuperar o segredo com `restore-secret` até a data e a hora presentes no campo de resposta `DeletionDate`. Para excluir um segredo que está replicado em outras regiões, primeiro remova suas réplicas com `remove-regions-from-replication` e então chame `delete-secret`.

```
aws secretsmanager delete-secret \  
  --secret-id MyTestSecret \  
  --recovery-window-in-days 7
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "DeletionDate": 1524085349.095  
}
```

Para obter mais informações, consulte [Excluir um segredo](#) no Guia do usuário do Secrets Manager.

Exemplo 2: como excluir um segredo instantaneamente

O exemplo de `delete-secret` a seguir exclui imediatamente um segredo sem uma janela de recuperação. Não é possível recuperar esse segredo.

```
aws secretsmanager delete-secret \  
  --secret-id MyTestSecret \  
  --force-delete-without-recovery
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "DeletionDate": 1508750180.309  
}
```

Para obter mais informações, consulte [Excluir um segredo](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [DeleteSecret](#) na Referência de AWS CLI Comandos.

describe-secret

O código de exemplo a seguir mostra como usar `describe-secret`.

AWS CLI

Como recuperar os detalhes de um segredo

O exemplo de `describe-secret` a seguir mostra os detalhes de um segredo.

```
aws secretsmanager describe-secret \  
--secret-id MyTestSecret
```

Saída:

```
{  
  "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"
    }
]
}

```

Para obter mais informações, consulte [Segredo](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [DescribeSecret](#) na Referência de AWS CLI Comandos.

get-random-password

O código de exemplo a seguir mostra como usar `get-random-password`.

AWS CLI

Para gerar uma senha aleatória

O `get-random-password` exemplo a seguir gera uma senha aleatória de 20 caracteres que inclui pelo menos uma letra maiúscula, minúscula, número e pontuação.

```

aws secretsmanager get-random-password \
  --require-each-included-type \

```

```
--password-length 20
```

Saída:

```
{
  "RandomPassword": "EXAMPLE-PASSWORD"
}
```

Para obter mais informações, consulte [Criar e gerenciar segredos](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [GetRandomPassword](#) na Referência de AWS CLI Comandos.

get-resource-policy

O código de exemplo a seguir mostra como usar `get-resource-policy`.

AWS CLI

Para recuperar a política baseada em recursos anexada a um segredo

O exemplo de `get-resource-policy` a seguir recupera a política baseada em recurso anexada a um segredo.

```
aws secretsmanager get-resource-policy \
  --secret-id MyTestSecret
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "ResourcePolicy": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n      \"Effect\":\n        \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam::123456789012:root\"\n      },\n      \"Action\":\n        \"secretsmanager:GetSecretValue\",\n      \"Resource\": \"*\"\n    }\n  ]\n}"
```

Para obter mais informações, consulte [Autenticação e controle de acesso](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [GetResourcePolicy](#) na Referência de AWS CLI Comandos.

get-secret-value

O código de exemplo a seguir mostra como usar `get-secret-value`.

AWS CLI

Exemplo 1: como recuperar o valor secreto criptografado de um segredo

O exemplo de `get-secret-value` a seguir obtém o valor atual do segredo.

```
aws secretsmanager get-secret-value \  
  --secret-id MyTestSecret
```

Saída:

```
{  
  "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  
}
```

Para obter mais informações, consulte [Recuperar segredos](#) no Guia do usuário do Secrets Manager.

Exemplo 2: como recuperar o valor secreto anterior

O seguinte exemplo de `get-secret-value` recupera o valor secreto anterior:

```
aws secretsmanager get-secret-value \  
  --secret-id MyTestSecret  
  --version-stage AWSPREVIOUS
```

Saída:


```
{
  "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
}
```

Para obter mais informações, consulte [Recuperar segredos](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [GetSecretValue](#) na Referência de AWS CLI Comandos.

list-secret-version-ids

O código de exemplo a seguir mostra como usar `list-secret-version-ids`.

AWS CLI

Para listar todas as versões secretas associadas a um segredo

O `list-secret-version-ids` exemplo a seguir obtém uma lista de todas as versões de um segredo.

```
aws secretsmanager list-secret-version-ids \
  --secret-id MyTestSecret
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Versão](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [ListSecretVersionIds](#) na Referência de AWS CLI Comandos.

list-secrets

O código de exemplo a seguir mostra como usar `list-secrets`.

AWS CLI

Exemplo 1: como listar os segredos na conta

O exemplo de `list-secrets` a seguir mostra uma lista dos segredos em sua conta.

```
aws secretsmanager list-secrets
```

Saída:

```
{
  "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"
      ]
    }
  }
]
}

```

Para obter mais informações, consulte [Localizar segredos](#) no Guia do usuário do Secrets Manager.

Exemplo 2: como filtrar os segredos na conta

O exemplo de `list-secrets` a seguir obtém uma lista dos segredos em sua conta que têm Test no nome. A filtragem por nome diferencia maiúsculas de minúsculas.

```

aws secretsmanager list-secrets \
  --filter Key="name",Values="Test"

```

Saída:

```

{
  "SecretList": [
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
      "Name": "MyTestSecret",
      "LastChangedDate": 1523477145.729,

```

```

        "SecretVersionsToStages": {
            "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
                "AWSCURRENT"
            ]
        }
    ]
}

```

Para obter mais informações, consulte [Localizar segredos](#) no Guia do usuário do Secrets Manager.

Exemplo 3: como listar os segredos na conta gerenciada por outro serviço

O `list-secrets` exemplo a seguir retorna os segredos da sua conta que são gerenciados pela AmazonRDS.

```

aws secretsmanager list-secrets \
  --filter Key="owning-service",Values="rds"

```

Saída:

```

{
  "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-EXAMPLEaaaa": [
        "AWSPREVIOUS"
      ],
      "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb": [
        "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"
  }
]
}

```

Para obter mais informações, consulte o [Segredos gerenciados por outros serviços](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [ListSecrets](#) na Referência de AWS CLI Comandos.

put-resource-policy

O código de exemplo a seguir mostra como usar `put-resource-policy`.

AWS CLI

Para adicionar uma política baseada em recursos a um segredo

O exemplo de `put-resource-policy` a seguir adiciona uma política de permissões a um segredo, verificando primeiro se a política não fornece acesso amplo ao segredo. A política é lida de um arquivo. Para obter mais informações, consulte [Carregando AWS CLI parâmetros de um arquivo](#) no Guia AWS CLI do usuário.

```

aws secretsmanager put-resource-policy \
  --secret-id MyTestSecret \
  --resource-policy file://mypolicy.json \

```

--block-public-policy

Conteúdo de mypolicy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/MyRole"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Para obter mais informações, consulte [Anexar uma política de permissões a um segredo](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [PutResourcePolicy](#) na Referência de AWS CLI Comandos.

put-secret-value

O código de exemplo a seguir mostra como usar `put-secret-value`.

AWS CLI

Exemplo 1: como armazenar um novo valor secreto em um segredo

O exemplo de `put-secret-value` a seguir cria uma nova versão de um segredo com dois pares de chave/valor.

```
aws secretsmanager put-secret-value \  
  --secret-id MyTestSecret \  
  --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "VersionStages": [  
    "AWSCURRENT"  
  ]  
}
```

Para obter mais informações, consulte [Modificar um segredo](#) no Guia do usuário do Secrets Manager.

Exemplo 2: Para armazenar um novo valor secreto das credenciais em um arquivo JSON

O exemplo de `put-secret-value` a seguir cria uma nova versão de um segredo com base em credenciais em um arquivo. Para obter mais informações, consulte [Carregando AWS CLI parâmetros de um arquivo](#) no Guia AWS CLI do usuário.

```
aws secretsmanager put-secret-value \  
  --secret-id MyTestSecret \  
  --secret-string file://mycreds.json
```

Conteúdo de `mycreds.json`:

```
{  
  "engine": "mysql",  
  "username": "saanvis",  
  "password": "EXAMPLE-PASSWORD",  
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",  
  "dbname": "myDatabase",  
  "port": "3306"  
}
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "VersionStages": [
    "AWSCURRENT"
  ]
}
```

Para obter mais informações, consulte [Modificar um segredo](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [PutSecretValue](#) na Referência de AWS CLI Comandos.

remove-regions-from-replication

O código de exemplo a seguir mostra como usar `remove-regions-from-replication`.

AWS CLI

Para excluir uma réplica secreta

O exemplo de `remove-regions-from-replication` a seguir exclui um segredo de réplica em `eu-west-3`. Para excluir um segredo primário que está replicado em outras regiões, primeiro remova as réplicas e então chame `delete-secret`.

```
aws secretsmanager remove-regions-from-replication \
  --secret-id MyTestSecret \
  --remove-replica-regions eu-west-3
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
  "ReplicationStatus": []
}
```

Para obter mais informações, consulte [Excluir uma réplica secreta](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [RemoveRegionsFromReplication](#) na Referência de AWS CLI Comandos.

replicate-secret-to-regions

O código de exemplo a seguir mostra como usar `replicate-secret-to-regions`.

AWS CLI

Para replicar um segredo para outra região

O exemplo de `replicate-secret-to-regions` a seguir replica um segredo para eu-west-3. A réplica é criptografada com a chave AWS `aws/secretsmanager` gerenciada.

```
aws secretsmanager replicate-secret-to-regions \  
  --secret-id MyTestSecret \  
  --add-replica-regions Region=eu-west-3
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "ReplicationStatus": [  
    {  
      "Region": "eu-west-3",  
      "KmsKeyId": "alias/aws/secretsmanager",  
      "Status": "InProgress"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Replicar um segredo para outra região](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [ReplicateSecretToRegions](#) na Referência de AWS CLI Comandos.

restore-secret

O código de exemplo a seguir mostra como usar `restore-secret`.

AWS CLI

Para restaurar um segredo excluído anteriormente

O exemplo de `restore-secret` a seguir restaura um segredo que estava previamente programado para exclusão.

```
aws secretsmanager restore-secret \  
  --secret-id MyTestSecret
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Para obter mais informações, consulte [Excluir um segredo](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [RestoreSecret](#) na Referência de AWS CLI Comandos.

rotate-secret

O código de exemplo a seguir mostra como usar `rotate-secret`.

AWS CLI

Exemplo 1: Para configurar e iniciar a rotação automática de um segredo

O `rotate-secret` exemplo a seguir configura e inicia a rotação automática de um segredo. O Secrets Manager gira o segredo uma vez imediatamente e depois a cada oito horas em uma janela de duas horas. A saída mostra a `VersionId` da nova versão secreta criada por rotação.

```
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\"}'"
```

Saída:

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Rotare secrets](#) no Guia do Usuário do Secrets Manager.

Exemplo 2: Para configurar e iniciar a rotação automática em um intervalo de rotação

O `rotate-secret` exemplo a seguir configura e inicia a rotação automática de um segredo. O Secrets Manager alterna o segredo uma vez imediatamente e depois a cada 10 dias. A saída mostra a `VersionId` da nova versão secreta criada por rotação.

```
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)\"}"
```

Saída:

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Rotare secrets](#) no Guia do Usuário do Secrets Manager.

Exemplo 3: Para girar um segredo imediatamente

O exemplo de `rotate-secret` a seguir inicia uma alternância imediata. A saída mostra a `VersionId` da nova versão secreta criada por rotação. O segredo já deve ter a alternância configurada.

```
aws secretsmanager rotate-secret \
```

```
--secret-id MyTestDatabaseSecret
```

Saída:

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Rotate secrets](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [RotateSecret](#) na Referência de AWS CLI Comandos.

stop-replication-to-replica

O código de exemplo a seguir mostra como usar stop-replication-to-replica.

AWS CLI

Para promover uma réplica secreta para uma primária

O exemplo de stop-replication-to-replica a seguir remove o link entre um segredo de réplica e o primário. O segredo de réplica é promovido a um segredo primário na região da réplica. É necessário chamar stop-replication-to-replica diretamente da região da réplica.

```
aws secretsmanager stop-replication-to-replica \
  --secret-id MyTestSecret
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3"
}
```

Para obter mais informações, consulte [Promover uma réplica secreta](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [StopReplicationToReplicana](#) Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Exemplo 1: Para adicionar uma tag a um segredo

O exemplo de a seguir mostra como anexar um tag com uma sintaxe abreviada.

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags Key=FirstTag,Value=FirstValue
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar seus segredos](#) no Guia do Usuário do Secrets Manager.

Exemplo 2: Para adicionar várias tags a um segredo

O exemplo de `tag-resource` a seguir anexa duas tags de chave/valor a um segredo.

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",  
  "Value": "SecondValue"}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Segredos de tags](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um segredo

O exemplo de `untag-resource` a seguir remove duas tags de um segredo. Para cada tag, tanto a chave quanto o valor são removidos.

```
aws secretsmanager untag-resource \  
  --secret-id MyTestSecret \  
  --tag-keys '[ "FirstTag", "SecondTag" ]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Segredos de tags](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-secret-version-stage

O código de exemplo a seguir mostra como usar `update-secret-version-stage`.

AWS CLI

Exemplo 1: Para reverter um segredo para a versão anterior

O `update-secret-version-stage` exemplo a seguir move o rótulo AWS CURRENT de teste para a versão anterior de um segredo, o que reverte o segredo para a versão anterior. Para encontrar o ID da versão anterior, use `list-secret-version-ids`. Neste exemplo, a versão com o rótulo é `a1b2c3d4-5678-90ab-cdef-` e a versão com o AWS CURRENT rótulo é `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`. Neste exemplo, você move o AWS CURRENT rótulo da versão `11111` para `22222`. Como o AWS CURRENT rótulo é removido de uma versão, `update-secret-version-stage` automaticamente o AWS PREVIOUS rótulo para essa versão (`11111`). O efeito é que as AWS PREVIOUS versões AWS CURRENT e são trocadas.

```
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
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Para obter mais informações, consulte [Versão](#) no Guia do Usuário do Secrets Manager.

Exemplo 2: Para adicionar uma etiqueta de teste anexada a uma versão de um segredo

O `update-secret-version-stage` exemplo a seguir adiciona um rótulo de teste a uma versão de um segredo. Você pode revisar os resultados executando `list-secret-version-ids` e visualizando o campo de `VersionStages` resposta da versão afetada.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage STAGINGLABEL1 \
  --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Para obter mais informações, consulte [Versão](#) no Guia do Usuário do Secrets Manager.

Exemplo 3: Para excluir uma etiqueta de teste anexada a uma versão de um segredo

O `update-secret-version-stage` exemplo a seguir exclui um rótulo de teste anexado a uma versão de um segredo. Você pode revisar os resultados executando `list-secret-version-ids` e visualizando o campo de `VersionStages` resposta da versão afetada.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage STAGINGLABEL1 \
```

```
--remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Para obter mais informações, consulte [Versão](#) no Guia do Usuário do Secrets Manager.

- Para API obter detalhes, consulte [UpdateSecretVersionStage](#) na Referência de AWS CLI Comandos.

update-secret

O código de exemplo a seguir mostra como usar update-secret.

AWS CLI

Exemplo 1: como atualizar a descrição de um segredo

O exemplo de update-secret a seguir retorna a descrição de um segredo.

```
aws secretsmanager update-secret \
  --secret-id MyTestSecret \
  --description "This is a new description for the secret."
```

Saída:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Para obter mais informações, consulte [Modificar um segredo](#) no Guia do usuário do Secrets Manager.

Exemplo 2: como atualizar a chave de criptografia associada a um segredo

O `update-secret` exemplo a seguir atualiza a KMS chave usada para criptografar o valor secreto. A KMS chave deve estar na mesma região do segredo.

```
aws secretsmanager update-secret \  
  --secret-id MyTestSecret \  
  --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-  
ba987EXAMPLE
```

Saída:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Para obter mais informações, consulte [Modificar um segredo](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [UpdateSecret](#) na Referência de AWS CLI Comandos.

validate-resource-policy

O código de exemplo a seguir mostra como usar `validate-resource-policy`.

AWS CLI

Para validar uma política de recursos

O `validate-resource-policy` exemplo a seguir verifica se uma política de recursos não concede amplo acesso a um segredo. A política é lida de um arquivo no disco. Para obter mais informações, consulte [Carregando AWS CLI parâmetros de um arquivo](#) no Guia AWS CLI do usuário.

```
aws secretsmanager validate-resource-policy \  
  --resource-policy file://mypolicy.json
```

Conteúdo de `mypolicy.json`:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:role/MyRole"
    },
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "*"
  }
]
```

Saída:

```
{
  "PolicyValidationPassed": true,
  "ValidationErrors": []
}
```

Para obter mais informações, consulte [Referência de permissões para o Secrets Manager](#) no Guia do usuário do Secrets Manager.

- Para API obter detalhes, consulte [ValidateResourcePolicy](#) na Referência de AWS CLI Comandos.

Exemplos do Security Hub usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Security Hub.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-administrator-invitation

O código de exemplo a seguir mostra como usar `accept-administrator-invitation`.

AWS CLI

Para aceitar um convite de uma conta de administrador

O `accept-administrator-invitation` exemplo a seguir aceita o convite especificado da conta de administrador especificada.

```
aws securityhub accept-invitation \  
  --administrator-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [AcceptAdministratorInvitation](#) na Referência de AWS CLI Comandos.

accept-invitation

O código de exemplo a seguir mostra como usar `accept-invitation`.

AWS CLI

Para aceitar um convite de uma conta de administrador

O `accept-invitation` exemplo a seguir aceita o convite especificado da conta de administrador especificada.

```
aws securityhub accept-invitation \  
  --master-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [AcceptInvitation](#) na Referência de AWS CLI Comandos.

batch-delete-automation-rules

O código de exemplo a seguir mostra como usar `batch-delete-automation-rules`.

AWS CLI

Para excluir regras de automação

O `batch-delete-automation-rules` exemplo a seguir exclui a regra de automação especificada. Você pode excluir uma ou mais regras com um único comando. Somente a conta de administrador do Security Hub pode executar esse comando.

```
aws securityhub batch-delete-automation-rules \
  --automation-rules-arns ['arn:aws:securityhub:us-
  east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111']
```

Saída:

```
{
  "ProcessedAutomationRules": [
    "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
    a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  ],
  "UnprocessedAutomationRules": []
}
```

Para obter mais informações, consulte [Excluindo regras de automação](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchDeleteAutomationRules](#) na Referência de AWS CLI Comandos.

batch-disable-standards

O código de exemplo a seguir mostra como usar `batch-disable-standards`.

AWS CLI

Para desativar um padrão

O `batch-disable-standards` exemplo a seguir desativa o padrão associado à assinatura ARN especificada.

```
aws securityhub batch-disable-standards \  
  --standards-subscription-arns "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Desabilitar ou habilitar um padrão de segurança](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchDisableStandards](#) na Referência de AWS CLI Comandos.

batch-enable-standards

O código de exemplo a seguir mostra como usar `batch-enable-standards`.

AWS CLI

Para habilitar um padrão

O `batch-enable-standards` exemplo a seguir ativa o PCI DSS padrão para a conta solicitante.

```
aws securityhub batch-enable-standards \
  --standards-subscription-requests '{"StandardsArn": "arn:aws:securityhub:us-
  west-1::standards/pci-dss/v/3.2.1"}'
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Desabilitar ou habilitar um padrão de segurança](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchEnableStandards](#) na Referência de AWS CLI Comandos.

batch-get-automation-rules

O código de exemplo a seguir mostra como usar batch-get-automation-rules.

AWS CLI

Para obter detalhes das regras de automação

O batch-get-automation-rules exemplo a seguir obtém detalhes da regra de automação especificada. Você pode obter detalhes de uma ou mais regras de automação com um único comando.

```
aws securityhub batch-get-automation-rules \
  --automation-rules-arns '['arn:aws:securityhub:us-
  east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111']'
```

Saída:

```
{
  "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": []
}

```

Para obter mais informações, consulte [Visualizando as regras de automação](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchGetAutomationRules](#) na Referência de AWS CLI Comandos.

batch-get-configuration-policy-associations

O código de exemplo a seguir mostra como usar `batch-get-configuration-policy-associations`.

AWS CLI

Para obter detalhes da associação de configuração para um lote de destinos

O `batch-get-configuration-policy-associations` exemplo a seguir recupera os detalhes da associação para os destinos especificados. Você pode fornecer a conta IDs, a unidade IDs organizacional ou o ID raiz do alvo.

```

aws securityhub batch-get-configuration-policy-associations \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'

```

Saída:


```
{
  "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."
}
```

Para obter mais informações, consulte [Visualizando as políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchGetConfigurationPolicyAssociations](#) na Referência de AWS CLI Comandos.

batch-get-security-controls

O código de exemplo a seguir mostra como usar batch-get-security-controls.

AWS CLI

Para obter detalhes do controle de segurança

O batch-get-security-controls exemplo a seguir obtém detalhes dos controles de segurança ACM .1 e IAM .1 na AWS conta atual e AWS na região.

```
aws securityhub batch-get-security-controls \
  --security-control-ids ['ACM.1', 'IAM.1']
```

Saída:

```
{
  "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": {}
}
]
}

```

Para obter mais informações, consulte [Visualizando detalhes de um controle](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchGetSecurityControls](#) na Referência de AWS CLI Comandos.

batch-get-standards-control-associations

O código de exemplo a seguir mostra como usar `batch-get-standards-control-associations`.

AWS CLI

Para obter o status de ativação de um controle

O `batch-get-standards-control-associations` exemplo a seguir identifica se os controles especificados estão habilitados nos padrões especificados.

```
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"}]'
```

Saída:

```
{
  "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"
    ]
}
]
}

```

Para obter mais informações, consulte [Habilitar e desabilitar controles em padrões específicos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchGetStandardsControlAssociations](#) na Referência de AWS CLI Comandos.

batch-import-findings

O código de exemplo a seguir mostra como usar batch-import-findings.

AWS CLI

Para atualizar uma descoberta

O `batch-import-findings` exemplo a seguir atualiza uma descoberta.

```
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"
    }]'
```

Saída:

```
{
  "FailedCount": 0,
  "SuccessCount": 1,
  "FailedFindings": []
}
```

Para obter mais informações, consulte [Usando BatchImportFindings para criar e atualizar descobertas](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchImportFindings](#) na Referência de AWS CLI Comandos.

batch-update-automation-rules

O código de exemplo a seguir mostra como usar batch-update-automation-rules.

AWS CLI

Para atualizar as regras de automação

O batch-update-automation-rules exemplo a seguir atualiza a regra de automação especificada. Você pode atualizar uma ou mais regras com um único comando. Somente a conta de administrador do Security Hub pode executar esse 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" \
  ] \
```

`]'`

Saída:

```
{
  "ProcessedAutomationRules": [
    "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  ],
  "UnprocessedAutomationRules": []
}
```

Para obter mais informações, consulte [Edição de regras de automação](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchUpdateAutomationRules](#) na Referência de AWS CLI Comandos.

batch-update-findings

O código de exemplo a seguir mostra como usar batch-update-findings.

AWS CLI

Exemplo 1: Para atualizar uma descoberta

O batch-update-findings exemplo a seguir atualiza duas descobertas para adicionar uma nota, alterar o rótulo de gravidade e resolvê-la.

```
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"}'
```

Saída:

```
{
  "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": []
}
```

Para obter mais informações, consulte [Usando BatchUpdateFindings para atualizar uma descoberta](#) no Guia do Usuário do AWS Security Hub.

Exemplo 2: Para atualizar uma descoberta usando a sintaxe abreviada

O batch-update-findings exemplo a seguir atualiza duas descobertas para adicionar uma nota, alterar o rótulo de gravidade e resolvê-la usando a sintaxe abreviada.

```
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"
```

Saída:

```
{
  "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": []
}

```

Para obter mais informações, consulte [Usando BatchUpdateFindings para atualizar uma descoberta](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchUpdateFindings](#) na Referência de AWS CLI Comandos.

batch-update-standards-control-associations

O código de exemplo a seguir mostra como usar batch-update-standards-control-associations.

AWS CLI

Para atualizar o status de habilitação de um controle em padrões habilitados

O batch-update-standards-control-associations exemplo a seguir desativa CloudTrail .1 nos padrões especificados.

```

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"}]'

```

Esse comando não gera nenhuma saída quando é bem-sucedido.

Para obter mais informações, consulte [Habilitar e desabilitar controles em padrões específicos](#) e [Habilitar e desabilitar controles em todos os padrões no Guia](#) do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [BatchUpdateStandardsControlAssociations](#) na Referência de AWS CLI Comandos.

create-action-target

O código de exemplo a seguir mostra como usar `create-action-target`.

AWS CLI

Para criar uma ação personalizada

O `create-action-target` exemplo a seguir cria uma ação personalizada. Ele fornece o nome, a descrição e o identificador da ação.

```
aws securityhub create-action-target \  
  --name "Send to remediation" \  
  --description "Action to send the finding for remediation tracking" \  
  --id "Remediation"
```

Saída:

```
{  
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation"  
}
```

Para obter mais informações, consulte [Criar uma ação personalizada e associá-la a uma regra de CloudWatch Eventos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [CreateActionTarget](#) na Referência de AWS CLI Comandos.

create-automation-rule

O código de exemplo a seguir mostra como usar `create-automation-rule`.

AWS CLI

Para criar uma regra de automação

O `create-automation-rule` exemplo a seguir cria uma regra de automação na AWS conta corrente e AWS na região. O Security Hub filtra suas descobertas com base nos critérios

especificados e aplica as ações às descobertas correspondentes. Somente a conta de administrador do Security Hub pode executar esse 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"
```

Saída:

```
{
  "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Criação de regras de automação](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [CreateAutomationRule](#) na Referência de AWS CLI Comandos.

create-configuration-policy

O código de exemplo a seguir mostra como usar `create-configuration-policy`.

AWS CLI

Para criar uma política de configuração

O `create-configuration-policy` exemplo a seguir cria uma política de configuração com as configurações especificadas.

```
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"}
```

Saída:

```
{
  "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
              }
            }
          }
        }
      ]
    }
  }
}

```

Para obter mais informações, consulte [Criação e associação de políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [CreateConfigurationPolicy](#) na Referência de AWS CLI Comandos.

create-finding-aggregator

O código de exemplo a seguir mostra como usar `create-finding-aggregator`.

AWS CLI

Para habilitar a agregação de localização

O `create-finding-aggregator` exemplo a seguir configura a agregação de localização. É administrado a partir do Leste dos EUA (Virgínia), que designa o Leste dos EUA (Virgínia) como a Região de agregação. Isso indica vincular somente regiões especificadas e não vincular automaticamente novas regiões. Ele seleciona o Oeste dos EUA (Norte da Califórnia) e o Oeste dos EUA (Oregon) como as regiões vinculadas.

```
aws securityhub create-finding-aggregator \  
  --region us-east-1 \  
  --region-linking-mode SPECIFIED_REGIONS \  
  --regions us-west-1,us-west-2
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Habilitando a agregação de localização](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [CreateFindingAggregator](#) na Referência de AWS CLI Comandos.

create-insight

O código de exemplo a seguir mostra como usar create-insight.

AWS CLI

Para criar uma visão personalizada

O create-insight exemplo a seguir cria uma visão personalizada chamada Descobertas críticas de funções, que retorna descobertas críticas relacionadas às AWS funções.

```
aws securityhub create-insight \  
  --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],  
  "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \  
  --group-by-attribute "ResourceId" \  
  --name "Critical role findings"
```

Saída:

```
{
```

```
"InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Para obter mais informações, consulte [Gerenciando insights personalizados](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [CreateInsight](#) na Referência de AWS CLI Comandos.

create-members

O código de exemplo a seguir mostra como usar `create-members`.

AWS CLI

Para adicionar contas como contas de membros

O `create-members` exemplo a seguir adiciona duas contas como contas de membro à conta do administrador solicitante.

```
aws securityhub create-members \
  --account-details '[{"AccountId": "123456789111"}, {"AccountId":
  "123456789222"}]'
```

Saída:

```
{
  "UnprocessedAccounts": []
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [CreateMembers](#) na Referência de AWS CLI Comandos.

decline-invitations

O código de exemplo a seguir mostra como usar `decline-invitations`.

AWS CLI

Para recusar um convite para ser uma conta de membro

O `decline-invitations` exemplo a seguir recusa um convite para ser uma conta membro da conta de administrador especificada. A conta do membro é a conta solicitante.

```
aws securityhub decline-invitations \  
  --account-ids "123456789012"
```

Saída:

```
{  
  "UnprocessedAccounts": []  
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeclineInvitations](#) na Referência de AWS CLI Comandos.

delete-action-target

O código de exemplo a seguir mostra como usar `delete-action-target`.

AWS CLI

Para excluir uma ação personalizada

O `delete-action-target` exemplo a seguir exclui a ação personalizada identificada pelo especificadoARN.

```
aws securityhub delete-action-target \  
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
  Remediation"
```

Saída:

```
{  
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
  Remediation"  
}
```

Para obter mais informações, consulte [Criar uma ação personalizada e associá-la a uma regra de CloudWatch Eventos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeleteActionTarget](#) na Referência de AWS CLI Comandos.

delete-configuration-policy

O código de exemplo a seguir mostra como usar `delete-configuration-policy`.

AWS CLI

Para excluir uma política de configuração

O `delete-configuration-policy` exemplo a seguir exclui a política de configuração especificada.

```
aws securityhub delete-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Este comando não produz saída.

Para obter mais informações, consulte [Excluindo e desassociando as políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeleteConfigurationPolicy](#) na Referência de AWS CLI Comandos.

delete-finding-aggregator

O código de exemplo a seguir mostra como usar `delete-finding-aggregator`.

AWS CLI

Para parar de encontrar agregação

O `delete-finding-aggregator` exemplo a seguir interrompe a localização da agregação. É administrado a partir do Leste dos EUA (Virgínia), que é a região de agregação.

```
aws securityhub delete-finding-aggregator \  
  --region us-east-1 \  
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-  
aggregator/123e4567-e89b-12d3-a456-426652340000
```

Este comando não produz saída.

Para obter mais informações, consulte Como [interromper a localização da agregação](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeleteFindingAggregator](#) na Referência de AWS CLI Comandos.

delete-insight

O código de exemplo a seguir mostra como usar delete-insight.

AWS CLI

Para excluir um insight personalizado

O delete-insight exemplo a seguir exclui o insight personalizado com o especificado ARN.

```
aws securityhub delete-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Saída:

```
{  
  "InsightArn": "arn:aws:securityhub:eu-  
  central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-  
  EXAMPLE11111"  
}
```

Para obter mais informações, consulte [Gerenciando insights personalizados](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeleteInsight](#) na Referência de AWS CLI Comandos.

delete-invitations

O código de exemplo a seguir mostra como usar delete-invitations.

AWS CLI

Para excluir um convite para ser uma conta de membro

O `delete-invitations` exemplo a seguir exclui um convite para ser uma conta de membro da conta de administrador especificada. A conta do membro é a conta solicitante.

```
aws securityhub delete-invitations \  
  --account-ids "123456789012"
```

Saída:

```
{  
  "UnprocessedAccounts": []  
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeleteInvitations](#) na Referência de AWS CLI Comandos.

delete-members

O código de exemplo a seguir mostra como usar `delete-members`.

AWS CLI

Para excluir contas de membros

O `delete-members` exemplo a seguir exclui as contas de membros especificadas da conta do administrador solicitante.

```
aws securityhub delete-members \  
  --account-ids "123456789111" "123456789222"
```

Saída:

```
{  
  "UnprocessedAccounts": []  
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DeleteMembers](#) na Referência de AWS CLI Comandos.

describe-action-targets

O código de exemplo a seguir mostra como usar `describe-action-targets`.

AWS CLI

Para recuperar detalhes sobre ações personalizadas

O `describe-action-targets` exemplo a seguir recupera informações sobre a ação personalizada identificada pelo especificadoARN.

```
aws securityhub describe-action-targets \
  --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/
  Remediation"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Criar uma ação personalizada e associá-la a uma regra de CloudWatch Eventos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DescribeActionTargets](#) na Referência de AWS CLI Comandos.

describe-hub

O código de exemplo a seguir mostra como usar `describe-hub`.

AWS CLI

Para obter informações sobre um recurso do hub

O `describe-hub` exemplo a seguir retorna a data de assinatura do recurso de hub especificado. O recurso do hub é identificado por seu ARN.

```
aws securityhub describe-hub \  
--hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Saída:

```
{  
  "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",  
  "SubscribedAt": "2019-11-19T23:15:10.046Z"  
}
```

Para obter mais informações, consulte [AWS::SecurityHub: :Hub](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [DescribeHub](#) na Referência de AWS CLI Comandos.

describe-organization-configuration

O código de exemplo a seguir mostra como usar `describe-organization-configuration`.

AWS CLI

Para ver como o Security Hub está configurado para uma organização

O `describe-organization-configuration` exemplo a seguir retorna informações sobre a forma como uma organização está configurada no Security Hub. Neste exemplo, a organização usa a configuração central. Somente a conta de administrador do Security Hub pode executar esse comando.

```
aws securityhub describe-organization-configuration
```

Saída:

```
{  
  "AutoEnable": false,  
  "MemberAccountLimitReached": false,  
  "AutoEnableStandards": "NONE",  
  "OrganizationConfiguration": {
```

```
    "ConfigurationType": "LOCAL",
    "Status": "ENABLED",
    "StatusMessage": "Central configuration has been enabled successfully"
  }
}
```

Para obter mais informações, consulte [Gerenciando contas com AWS Organizations](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DescribeOrganizationConfiguration](#) na Referência de AWS CLI Comandos.

describe-products

O código de exemplo a seguir mostra como usar describe-products.

AWS CLI

Para retornar informações sobre as integrações de produtos disponíveis

O describe-products exemplo a seguir retorna as integrações de produtos disponíveis, uma por vez.

```
aws securityhub describe-products \
  --max-results 1
```

Saída:

```
{
  "NextToken": "U2FsdGVkX18vvP10qb7RD1rWRWVFBJI46M0IAb+nZmRJmR15NoRi2gm13sdQEn30/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\"}}}]}"
    }
  ]
}

```

Para obter mais informações, consulte [Gerenciando integrações de produtos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DescribeProducts](#) na Referência de AWS CLI Comandos.

describe-standards-controls

O código de exemplo a seguir mostra como usar `describe-standards-controls`.

AWS CLI

Para solicitar a lista de controles em um padrão habilitado

O `describe-standards-controls` exemplo a seguir solicita a lista de controles na assinatura padrão da conta do solicitante. PCI DSS A solicitação retorna dois controles por vez.

```
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
```

Saída:

```
{
  "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+eNkPoZHV111ip5HUYQPWSWZGmftcmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveo0dvfxXck0BaGhohIXhsIedN+LSjQV/
17kfCfJcq4PziNC1N9xe9aq2pjlLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUVoeriHZhyIiBgWKf7oNYdBVG80EortVWvSkoUTt
+B2ThcnC7143kI0UNx1kZ6sc64AsW"
}

```

Para obter mais informações, consulte [Visualizando detalhes dos controles](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DescribeStandardsControls](#) na Referência de AWS CLI Comandos.

describe-standards

O código de exemplo a seguir mostra como usar `describe-standards`.

AWS CLI

Para retornar uma lista dos padrões disponíveis

O `describe-standards` exemplo a seguir retorna a lista de padrões disponíveis.

```
aws securityhub describe-standards
```

Saída:

```

{
  "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
  }
]
}

```

Para obter mais informações, consulte [Padrões de segurança no AWS Security Hub](#) no Guia do usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DescribeStandards](#) na Referência de AWS CLI Comandos.

disable-import-findings-for-product

O código de exemplo a seguir mostra como usar `disable-import-findings-for-product`.

AWS CLI

Para parar de receber descobertas de uma integração de produto

O `disable-import-findings-for-product` exemplo a seguir desativa o fluxo de descobertas para a assinatura especificada de uma integração de produto.

```
aws securityhub disable-import-findings-for-product \  
  --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-  
  subscription/crowdstrike/crowdstrike-falcon"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciando integrações de produtos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DisableImportFindingsForProduct](#) na Referência de AWS CLI Comandos.

disable-organization-admin-account

O código de exemplo a seguir mostra como usar `disable-organization-admin-account`.

AWS CLI

Para remover uma conta de administrador do Security Hub

O `disable-organization-admin-account` exemplo a seguir revoga a atribuição da conta especificada como conta de administrador do Security Hub para Organizations AWS .

```
aws securityhub disable-organization-admin-account \  
  --admin-account-id 777788889999
```

Este comando não produz saída.

Para obter mais informações, consulte [Designação de uma conta de administrador do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DisableOrganizationAdminAccount](#) na Referência de AWS CLI Comandos.

disable-security-hub

O código de exemplo a seguir mostra como usar `disable-security-hub`.

AWS CLI

Para desativar o AWS Security Hub

O `disable-security-hub` exemplo a seguir desativa o AWS Security Hub para a conta solicitante.

```
aws securityhub disable-security-hub
```

Este comando não produz saída.

Para obter mais informações, consulte [Desabilitando o AWS Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DisableSecurityHub](#) na Referência de AWS CLI Comandos.

disassociate-from-administrator-account

O código de exemplo a seguir mostra como usar `disassociate-from-administrator-account`.

AWS CLI

Para se desassociar de uma conta de administrador

O `disassociate-from-administrator-account` exemplo a seguir desassocia a conta solicitante de sua conta de administrador atual.

```
aws securityhub disassociate-from-administrator-account
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DisassociateFromAdministratorAccount](#) na Referência de AWS CLI Comandos.

disassociate-from-master-account

O código de exemplo a seguir mostra como usar `disassociate-from-master-account`.

AWS CLI

Para se desassociar de uma conta de administrador

O `disassociate-from-master-account` exemplo a seguir desassocia a conta solicitante de sua conta de administrador atual.

```
aws securityhub disassociate-from-master-account
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DisassociateFromMasterAccount](#) na Referência de AWS CLI Comandos.

disassociate-members

O código de exemplo a seguir mostra como usar `disassociate-members`.

AWS CLI

Para desassociar contas-membro

O `disassociate-members` exemplo a seguir desassocia as contas de membros especificadas da conta de administrador solicitante.

```
aws securityhub disassociate-members \
  --account-ids "123456789111" "123456789222"
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [DisassociateMembers](#) na Referência de AWS CLI Comandos.

enable-import-findings-for-product

O código de exemplo a seguir mostra como usar `enable-import-findings-for-product`.

AWS CLI

Para começar a receber descobertas de uma integração de produtos

O `enable-import-findings-for-product` exemplo a seguir permite o fluxo de descobertas da integração de produtos especificada.

```
aws securityhub enable-import-findings-for-product \  
  --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/  
crowdstrike-falcon"
```

Saída:

```
{  
  "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-  
subscription/crowdstrike/crowdstrike-falcon"  
}
```

Para obter mais informações, consulte [Gerenciando integrações de produtos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [EnableImportFindingsForProduct](#) na Referência de AWS CLI Comandos.

enable-organization-admin-account

O código de exemplo a seguir mostra como usar `enable-organization-admin-account`.

AWS CLI

Para designar uma conta da organização como uma conta de administrador do Security Hub

O `enable-organization-admin-account` exemplo a seguir designa a conta especificada como uma conta de administrador do Security Hub.

```
aws securityhub enable-organization-admin-account \  
  --admin-account-id 777788889999
```

Este comando não produz saída.

Para obter mais informações, consulte [Designação de uma conta de administrador do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [EnableOrganizationAdminAccount](#) na Referência de AWS CLI Comandos.

enable-security-hub

O código de exemplo a seguir mostra como usar `enable-security-hub`.

AWS CLI

Para habilitar o AWS Security Hub

O `enable-security-hub` exemplo a seguir ativa o AWS Security Hub para a conta solicitante. Ele configura o Security Hub para habilitar os padrões padrão. Para o recurso do hub, ele atribui o valor `Security` à `tagDepartment`.

```
aws securityhub enable-security-hub \  
  --enable-default-standards \  
  --tags '{"Department": "Security"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Habilitando o Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [EnableSecurityHub](#) na Referência de AWS CLI Comandos.

get-administrator-account

O código de exemplo a seguir mostra como usar `get-administrator-account`.

AWS CLI

Para recuperar informações sobre uma conta de administrador

O `get-administrator-account` exemplo a seguir recupera informações sobre a conta do administrador da conta solicitante.

```
aws securityhub get-administrator-account
```

Saída:

```
{  
  "Master": {  
    "AccountId": "123456789012",  
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",  
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
```

```
    "MemberStatus": "ASSOCIATED"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetAdministratorAccount](#) na Referência de AWS CLI Comandos.

get-configuration-policy-association

O código de exemplo a seguir mostra como usar `get-configuration-policy-association`.

AWS CLI

Para obter detalhes da associação de configuração para um alvo

O `get-configuration-policy-association` exemplo a seguir recupera os detalhes da associação para o destino especificado. Você pode fornecer um ID da conta, ID da unidade organizacional ou ID raiz do alvo.

```
aws securityhub get-configuration-policy-association \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

Saída:

```
{
  "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."
}
```

Para obter mais informações, consulte [Visualizando as políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetConfigurationPolicyAssociation](#) na Referência de AWS CLI Comandos.

get-configuration-policy

O código de exemplo a seguir mostra como usar `get-configuration-policy`.

AWS CLI

Para ver os detalhes da política de configuração

O `get-configuration-policy` exemplo a seguir recupera detalhes sobre a política de configuração especificada.

```
aws securityhub get-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Saída:

```
{  
  "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": {
```



```
    "StandardsStatus": "READY",
    "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
  }
]
}
```

Para obter mais informações, consulte [Padrões de segurança no AWS Security Hub](#) no Guia do usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetEnabledStandards](#) na Referência de AWS CLI Comandos.

get-finding-aggregator

O código de exemplo a seguir mostra como usar `get-finding-aggregator`.

AWS CLI

Para recuperar a configuração atual de agregação de descoberta

O `get-finding-aggregator` exemplo a seguir recupera a configuração atual da agregação de descoberta.

```
aws securityhub get-finding-aggregator \
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Visualizando a configuração atual da agregação de descobertas](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetFindingAggregator](#) na Referência de AWS CLI Comandos.

get-finding-history

O código de exemplo a seguir mostra como usar `get-finding-history`.

AWS CLI

Para obter o histórico de descoberta

O `get-finding-history` exemplo a seguir mostra os últimos 90 dias do histórico da descoberta especificada. Neste exemplo, os resultados estão limitados a dois registros do histórico de descobertas.

```
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"
```

Saída:

```
{
  "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": []
}
]
}

```

Para obter mais informações, consulte [Encontrando o histórico](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetFindingHistory](#) na Referência de AWS CLI Comandos.

get-findings

O código de exemplo a seguir mostra como usar `get-findings`.

AWS CLI

Exemplo 1: Para retornar descobertas geradas para um padrão específico

O `get-findings` exemplo a seguir retorna as descobertas do PCI DSS padrão.

```
aws securityhub get-findings \
  --filters '{"GeneratorId":[{"Value": "pci-dss", "Comparison": "PREFIX"}]}' \
  --max-items 1
```

Saída:

```
{
  "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=="
}

```

Exemplo 2: Para retornar descobertas de gravidade crítica que tenham um status de fluxo de trabalho de NOTIFIED

O `get-findings` exemplo a seguir retorna descobertas que têm um valor de rótulo de gravidade CRITICAL e um status de fluxo de trabalho de NOTIFIED. Os resultados são classificados em ordem decrescente pelo valor de Confiança.

```

aws securityhub get-findings \
  --filters '{"SeverityLabel":[{"Value":
"CRITICAL", "Comparison": "EQUALS"}], "WorkflowStatus":
[{"Value": "NOTIFIED", "Comparison": "EQUALS"}]}' \
  --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
  --max-items 1

```

Saída:

```

{
  "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"
}
]
}

```

Para obter mais informações, consulte [Filtragem e agrupamento de descobertas no Guia do Usuário do AWS Security Hub](#).

- Para API obter detalhes, consulte [GetFindings](#) na Referência de AWS CLI Comandos.

get-insight-results

O código de exemplo a seguir mostra como usar `get-insight-results`.

AWS CLI

Para recuperar os resultados para obter uma visão

O `get-insight-results` exemplo a seguir retorna a lista de resultados do insight com o especificadoARN.

```
aws securityhub get-insight-results \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Para obter mais informações, consulte [Visualizando e tomando medidas sobre os resultados e descobertas do insight](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetInsightResults](#) na Referência de AWS CLI Comandos.

get-insights

O código de exemplo a seguir mostra como usar `get-insights`.

AWS CLI

Para recuperar detalhes sobre um insight

O `get-insights` exemplo a seguir recupera os detalhes da configuração do insight com o especificadoARN.

```
aws securityhub get-insights \
  --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Insights no AWS Security Hub](#) no Guia do usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetInsights](#) na Referência de AWS CLI Comandos.

get-invitations-count

O código de exemplo a seguir mostra como usar `get-invitations-count`.

AWS CLI

Para recuperar o número de convites que não foram aceitos

O `get-invitations-count` exemplo a seguir recupera o número de convites que a conta solicitante recusou ou não respondeu.

```
aws securityhub get-invitations-count
```

Saída:

```
{
  "InvitationsCount": 3
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetInvitationsCount](#) na Referência de AWS CLI Comandos.

get-master-account

O código de exemplo a seguir mostra como usar `get-master-account`.

AWS CLI

Para recuperar informações sobre uma conta de administrador

O `get-master-account` exemplo a seguir recupera informações sobre a conta do administrador da conta solicitante.

```
aws securityhub get-master-account
```

Saída:

```
{
  "Master": {
    "AccountId": "123456789012",
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
    "MemberStatus": "ASSOCIATED"
  }
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetMasterAccount](#) na Referência de AWS CLI Comandos.

get-members

O código de exemplo a seguir mostra como usar `get-members`.

AWS CLI

Para recuperar informações sobre contas de membros selecionadas

O `get-members` exemplo a seguir recupera informações sobre as contas de membros especificadas.

```
aws securityhub get-members \
  --account-ids "444455556666" "777788889999"
```

Saída:

```
{
  "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": [ ]
}

```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [GetMembers](#) na Referência de AWS CLI Comandos.

get-security-control-definition

O código de exemplo a seguir mostra como usar `get-security-control-definition`.

AWS CLI

Para obter detalhes da definição do controle de segurança

O `get-security-control-definition` exemplo a seguir recupera os detalhes da definição de um controle de segurança do Security Hub. Os detalhes incluem o título do controle, a descrição, a disponibilidade da região, os parâmetros e outras informações.

```
aws securityhub get-security-control-definition \
  --security-control-id ACM.1
```

Saída:

```

{
  "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

```


Saída:

```
{
  "UnprocessedAccounts": []
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [InviteMembers](#) na Referência de AWS CLI Comandos.

list-automation-rules

O código de exemplo a seguir mostra como usar `list-automation-rules`.

AWS CLI

Para ver uma lista de regras de automação

O `list-automation-rules` exemplo a seguir lista as regras de automação de uma AWS conta. Somente a conta de administrador do Security Hub pode executar esse comando.

```
aws securityhub list-automation-rules \
  --max-results 3 \
  --next-token NULL
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Visualizando as regras de automação](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListAutomationRules](#) na Referência de AWS CLI Comandos.

list-configuration-policies

O código de exemplo a seguir mostra como usar `list-configuration-policies`.

AWS CLI

Para listar resumos de políticas de configuração

O `list-configuration-policies` exemplo a seguir lista um resumo das políticas de configuração da organização.

```
aws securityhub list-configuration-policies \  
--max-items 3
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Visualizando as políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListConfigurationPolicies](#) na Referência de AWS CLI Comandos.

list-configuration-policy-associations

O código de exemplo a seguir mostra como usar `list-configuration-policy-associations`.

AWS CLI

Para listar associações de configuração

O `list-configuration-policy-associations` exemplo a seguir lista um resumo das associações de configuração da organização. A resposta inclui associações com políticas de configuração e comportamento autogerenciado.

```
aws securityhub list-configuration-policy-associations \
  --association-type "APPLIED" \
  --max-items 4
```

Saída:

```
{
  "ConfigurationPolicyAssociationSummaries": [
    {
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TargetId": "r-1ab2",
      "TargetType": "ROOT",
      "AssociationType": "APPLIED",
      "UpdatedAt": "2023-11-28T19:26:49.417000+00:00",
      "AssociationStatus": "FAILED",
      "AssociationStatusMessage": "Policy association failed because 2
organizational units or accounts under this root failed."
    },
    {
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "TargetId": "ou-1ab2-c3de4f5g",
      "TargetType": "ORGANIZATIONAL_UNIT",
      "AssociationType": "APPLIED",
      "UpdatedAt": "2023-09-26T21:14:05.283000+00:00",
      "AssociationStatus": "FAILED",
      "AssociationStatusMessage": "One or more children under this target
failed association."
    },
    {
      "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
      "TargetId": "ou-6hi7-8j91kl2m",
```

```
    "TargetType": "ORGANIZATIONAL_UNIT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
    "AssociationStatus": "SUCCESS",
    "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"
  }
}
```

Para obter mais informações, consulte [Visualizando as políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListConfigurationPolicyAssociations](#) na Referência de AWS CLI Comandos.

list-enabled-products-for-import

O código de exemplo a seguir mostra como usar `list-enabled-products-for-import`.

AWS CLI

Para retornar a lista de integrações de produtos ativadas

O `list-enabled-products-for-import` exemplo a seguir retorna a lista de assinaturas ARNS das integrações de produtos atualmente habilitadas.

```
aws securityhub list-enabled-products-for-import
```

Saída:

```
{
  "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-
west-1:123456789012:product-subscription/aws/securityhub" ]
}
```

```
}
```

Para obter mais informações, consulte [Gerenciando integrações de produtos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListEnabledProductsForImport](#) na Referência de AWS CLI Comandos.

list-finding-aggregators

O código de exemplo a seguir mostra como usar `list-finding-aggregators`.

AWS CLI

Para listar os widgets disponíveis

O `list-finding-aggregators` exemplo a seguir retorna a configuração ARN da agregação de descoberta.

```
aws securityhub list-finding-aggregators
```

Saída:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000"
}
```

Para obter mais informações, consulte [Visualizando a configuração atual da agregação de descobertas](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListFindingAggregators](#) na Referência de AWS CLI Comandos.

list-invitations

O código de exemplo a seguir mostra como usar `list-invitations`.

AWS CLI

Para exibir uma lista de convites

O `list-invitations` exemplo a seguir recupera a lista de convites enviados para a conta solicitante.

```
aws securityhub list-invitations
```

Saída:

```
{
  "Invitations": [
    {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
    }
  ],
}
```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListInvitations](#) na Referência de AWS CLI Comandos.

list-members

O código de exemplo a seguir mostra como usar `list-members`.

AWS CLI

Para recuperar uma lista de contas de membros

O `list-members` exemplo a seguir retorna a lista de contas de membros da conta de administrador solicitante.

```
aws securityhub list-members
```

Saída:

```
{
  "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
  }
],
}

```

Para obter mais informações, consulte [Gerenciamento de contas de administradores e membros](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListMembers](#) na Referência de AWS CLI Comandos.

list-organization-admin-accounts

O código de exemplo a seguir mostra como usar `list-organization-admin-accounts`.

AWS CLI

Para listar as contas de administrador do Security Hub designadas

O `list-organization-admin-accounts` exemplo a seguir lista as contas de administrador do Security Hub de uma organização.

```
aws securityhub list-organization-admin-accounts
```

Saída:

```

{
  AdminAccounts": [
    { "AccountId": "777788889999" },
    { "Status": "ENABLED" }
  ]
}

```



```
]
}
```

Para obter mais informações, consulte [Designação de uma conta de administrador do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListOrganizationAdminAccounts](#) na Referência de AWS CLI Comandos.

list-security-control-definitions

O código de exemplo a seguir mostra como usar `list-security-control-definitions`.

AWS CLI

Exemplo 1: Para listar todos os controles de segurança disponíveis

O `list-security-control-definitions` exemplo a seguir lista os controles de segurança disponíveis em todos os padrões do Security Hub. Este exemplo limita os resultados a três controles.

```
aws securityhub list-security-control-definitions \
  --max-items 3
```

Saída:

```
{
  "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=="
}

```

Para obter mais informações, consulte [Visualizando detalhes de um padrão](#) no Guia do Usuário do AWS Security Hub.

Exemplo 2: Para listar os controles de segurança disponíveis para um padrão específico

O `list-security-control-definitions` exemplo a seguir lista os controles de segurança disponíveis para o CIS AWS Foundations Benchmark v1.4.0. Este exemplo limita os resultados a três controles.

```
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
```

Saída:

```
{  
  "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=="
}

```

Para obter mais informações, consulte [Visualizando detalhes de um padrão](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListSecurityControlDefinitions](#) na Referência de AWS CLI Comandos.

list-standards-control-associations

O código de exemplo a seguir mostra como usar `list-standards-control-associations`.

AWS CLI

Para obter o status de ativação de um controle em cada padrão habilitado

O `list-standards-control-associations` exemplo a seguir lista o status de habilitação de CloudTrail .1 em cada padrão habilitado.

```
aws securityhub list-standards-control-associations \
--security-control-id CloudTrail.1
```

Saída:

```

{
  "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)."
}
]
}
```

Para obter mais informações, consulte [Habilitar e desabilitar controles em padrões específicos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [ListStandardsControlAssociations](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para recuperar as tags atribuídas a um recurso

O `list-tags-for-resource` exemplo a seguir retorna as tags atribuídas ao recurso de hub especificado.

```
aws securityhub list-tags-for-resource \
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Saída:

```
{
  "Tags": {
    "Department" : "Operations",
    "Area" : "USMidwest"
  }
}
```

Para obter mais informações, consulte [AWS::SecurityHub: :Hub](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

start-configuration-policy-association

O código de exemplo a seguir mostra como usar `start-configuration-policy-association`.

AWS CLI

Exemplo 1: Para associar uma política de configuração

O `start-configuration-policy-association` exemplo a seguir associa a política de configuração especificada à unidade organizacional especificada. Uma configuração pode estar associada a uma conta de destino, unidade organizacional ou raiz.

```
aws securityhub start-configuration-policy-association \  
  --configuration-policy-identifier "arn:aws:securityhub:eu-  
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

Saída:

```
{  
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
  "TargetId": "ou-6hi7-8j91kl2m",  
  "TargetType": "ORGANIZATIONAL_UNIT",  
  "AssociationType": "APPLIED",  
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",  
  "AssociationStatus": "PENDING"  
}
```

Para obter mais informações, consulte [Criação e associação de políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

Exemplo 2: Para associar uma configuração autogerenciada

O `start-configuration-policy-association` exemplo a seguir associa uma configuração autogerenciada à conta especificada.

```
aws securityhub start-configuration-policy-association \  
  --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \  
  --target '{"OrganizationalUnitId": "123456789012"}'
```

Saída:

```
{  
  "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",  
  "TargetId": "123456789012",  
  "TargetType": "ACCOUNT",  
  "AssociationType": "APPLIED",  
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",  
  "AssociationStatus": "PENDING"  
}
```

Para obter mais informações, consulte [Criação e associação de políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [StartConfigurationPolicyAssociation](#) na Referência de AWS CLI Comandos.

start-configuration-policy-disassociation

O código de exemplo a seguir mostra como usar `start-configuration-policy-disassociation`.

AWS CLI

Exemplo 1: Para desassociar uma política de configuração

O `start-configuration-policy-disassociation` exemplo a seguir dissocia uma política de configuração da unidade organizacional especificada. Uma configuração pode ser desassociada de uma conta de destino, unidade organizacional ou raiz.

```
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"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Desassociando uma configuração das contas e OUs no Guia do Usuário do AWS Security Hub](#).

Exemplo 2: Para desassociar uma configuração autogerenciada

O `start-configuration-policy-disassociation` exemplo a seguir desassocia uma configuração autogerenciada da conta especificada.

```
aws securityhub start-configuration-policy-disassociation \  
  --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \  
  --target '{"AccountId": "123456789012"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Desassociando uma configuração das contas e OUs no Guia do Usuário do AWS Security Hub](#).

- Para API obter detalhes, consulte [StartConfigurationPolicyDisassociation](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para atribuir uma tag a um recurso

O `tag-resource` exemplo a seguir atribui valores para as tags Departamento e Área ao recurso de hub especificado.

```
aws securityhub tag-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
  --tags '{"Department":"Operations", "Area":"USMidwest"}'
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS::SecurityHub: :Hub](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover um valor de tag de um recurso

O `untag-resource` exemplo a seguir remove a tag Department do recurso de hub especificado.

```
aws securityhub untag-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
  --tag-keys "Department"
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS::SecurityHub: :Hub](#) no Guia do AWS CloudFormation Usuário.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-action-target

O código de exemplo a seguir mostra como usar `update-action-target`.

AWS CLI

Para atualizar uma ação personalizada

O `update-action-target` exemplo a seguir atualiza o nome da ação personalizada identificada pelo especificadoARN.

```
aws securityhub update-action-target \  
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation" \  
  --name "Send to remediation"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar uma ação personalizada e associá-la a uma regra de CloudWatch Eventos](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateActionTarget](#) na Referência de AWS CLI Comandos.

update-configuration-policy

O código de exemplo a seguir mostra como usar `update-configuration-policy`.

AWS CLI

Para atualizar uma política de configuração

O `update-configuration-policy` exemplo a seguir atualiza uma política de configuração existente para usar as configurações especificadas.

```
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"
```

Saída:

```
{
  "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
                }
              }
            }
          }
        ]
      }
    }
  }
}
```

```
    }  
  }  
}
```

Para obter mais informações, consulte [Atualização das políticas de configuração do Security Hub](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateConfigurationPolicy](#) na Referência de AWS CLI Comandos.

update-finding-aggregator

O código de exemplo a seguir mostra como usar `update-finding-aggregator`.

AWS CLI

Para atualizar a configuração atual da agregação de descoberta

O `update-finding-aggregator` exemplo a seguir altera a configuração da agregação de localização para vincular a partir de regiões selecionadas. É administrado a partir do Leste dos EUA (Virgínia), que é a região de agregação. Ele seleciona o Oeste dos EUA (Norte da Califórnia) e o Oeste dos EUA (Oregon) como as regiões vinculadas.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Atualização da configuração de agregação de localização](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateFindingAggregator](#) na Referência de AWS CLI Comandos.

update-insight

O código de exemplo a seguir mostra como usar `update-insight`.

AWS CLI

Exemplo 1: Para alterar o filtro de um insight personalizado

O `update-insight` exemplo a seguir altera os filtros de uma visão personalizada. O insight atualizado busca descobertas com alta severidade relacionadas às AWS funções.

```
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"
```

Exemplo 2: Para alterar o atributo de agrupamento para um insight personalizado

O `update-insight` exemplo a seguir altera o atributo de agrupamento do insight personalizado com o especificado ARN. O novo atributo de agrupamento é o ID do recurso.

```
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"
```

Saída:

```
{
  "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"
    }
]
}

```

Para obter mais informações, consulte [Gerenciando insights personalizados](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateInsight](#) na Referência de AWS CLI Comandos.

update-organization-configuration

O código de exemplo a seguir mostra como usar `update-organization-configuration`.

AWS CLI

Para atualizar a forma como o Security Hub está configurado para uma organização

O `update-organization-configuration` exemplo a seguir especifica que o Security Hub deve usar a configuração central para configurar uma organização. Depois de executar esse comando, o administrador delegado do Security Hub pode criar e gerenciar políticas de configuração para configurar a organização. O administrador delegado também pode usar esse comando para alternar da configuração central para a local. Se a configuração local for o tipo de configuração, o administrador delegado poderá escolher se deseja ativar automaticamente o Security Hub e os padrões de segurança padrão nas novas contas da organização.

```

aws securityhub update-organization-configuration \
  --no-auto-enable \
  --organization-configuration '{"ConfigurationType": "CENTRAL"}'

```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciando contas com AWS Organizations](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateOrganizationConfiguration](#) na Referência de AWS CLI Comandos.

update-security-control

O código de exemplo a seguir mostra como usar `update-security-control`.

AWS CLI

Para atualizar as propriedades do controle de segurança

O `update-security-control` exemplo a seguir especifica valores personalizados para um parâmetro de controle de segurança do 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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Parâmetros de controle personalizados](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateSecurityControl](#) na Referência de AWS CLI Comandos.

update-security-hub-configuration

O código de exemplo a seguir mostra como usar `update-security-hub-configuration`.

AWS CLI

Para atualizar a configuração do Security Hub

O `update-security-hub-configuration` exemplo a seguir configura o Security Hub para habilitar automaticamente novos controles para padrões habilitados.

```
aws securityhub update-security-hub-configuration \  
  --auto-enable-controls
```


Este comando não produz saída.

Para obter mais informações, consulte [Habilitando novos controles automaticamente](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateSecurityHubConfiguration](#) na Referência de AWS CLI Comandos.

update-standards-control

O código de exemplo a seguir mostra como usar `update-standards-control`.

AWS CLI

Exemplo 1: Para desativar um controle

O `update-standards-control` exemplo a seguir desativa o PCI AutoScaling1. controle.

```
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"
```

Este comando não produz saída.

Exemplo 2: Para ativar um controle

O `update-standards-control` exemplo a seguir habilita PCI o. AutoScaling1. controle.

```
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"
```

Este comando não produz saída.

Para obter mais informações, consulte [Desabilitar e ativar controles individuais](#) no Guia do Usuário do AWS Security Hub.

- Para API obter detalhes, consulte [UpdateStandardsControl](#) na Referência de AWS CLI Comandos.

AWS Serverless Application Repository exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Serverless Application Repository.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

put-application-policy

O código de exemplo a seguir mostra como usar `put-application-policy`.

AWS CLI

Exemplo 1: Para compartilhar um aplicativo publicamente

O seguinte `put-application-policy` compartilha um aplicativo publicamente, para que qualquer pessoa possa encontrar e implantar seu aplicativo no 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
```

Saída:

```
{  
  "Statements": [  
    {
```

```

    "Actions": [
      "Deploy"
    ],
    "Principals": [
      ""
    ],
    "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
]
}

```

Exemplo 2: Para compartilhar um aplicativo de forma privada

O seguinte `put-application-policy` compartilha um aplicativo de forma privada, para que somente AWS contas específicas possam encontrar e implantar seu aplicativo no 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

```

Saída:

```

{
  "Statements": [
    {
      "Actions": [
        "Deploy"
      ],
      "Principals": [
        "111111111111",
        "222222222222"
      ],
      "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
  ]
}

```

Para obter mais informações, consulte [Compartilhando um aplicativo por meio do console](#) no AWS Serverless Application Repository Developer Guide

- Para API obter detalhes, consulte [PutApplicationPolicy](#) na Referência de AWS CLI Comandos.

Exemplos do Service Catalog usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Service Catalog.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

accept-portfolio-share

O código de exemplo a seguir mostra como usar `accept-portfolio-share`.

AWS CLI

Para aceitar um compartilhamento de portfólio

O `accept-portfolio-share` exemplo a seguir aceita uma oferta, feita por outro usuário, para compartilhar o portfólio especificado.

```
aws servicecatalog accept-portfolio-share \  
--portfolio-id port-2s6wuabcdefghijk
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AcceptPortfolioShare](#) na Referência de AWS CLI Comandos.

associate-principal-with-portfolio

O código de exemplo a seguir mostra como usar `associate-principal-with-portfolio`.

AWS CLI

Para associar um diretor a um portfólio

O `associate-principal-with-portfolio` exemplo a seguir associa um usuário ao portfólio especificado.

```
aws servicecatalog associate-principal-with-portfolio \  
  --portfolio-id port-2s6abcdefwdh4 \  
  --principal-arn arn:aws:iam::123456789012:user/usertest \  
  --principal-type IAM
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AssociatePrincipalWithPortfolio](#) na Referência de AWS CLI Comandos.

`associate-product-with-portfolio`

O código de exemplo a seguir mostra como usar `associate-product-with-portfolio`.

AWS CLI

Para associar um produto a um portfólio

O `associate-product-with-portfolio` exemplo a seguir associa o produto fornecido ao portfólio especificado.

```
aws servicecatalog associate-product-with-portfolio \  
  --product-id prod-3p5abcdef3oyk \  
  --portfolio-id port-2s6abcdef5wdh4
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AssociateProductWithPortfolio](#) na Referência de AWS CLI Comandos.

`associate-tag-option-with-resource`

O código de exemplo a seguir mostra como usar `associate-tag-option-with-resource`.

AWS CLI

Para associar um TagOption a a um recurso

O `associate-tag-option-with-resource` exemplo a seguir associa o especificado TagOption ao recurso especificado.

```
aws servicecatalog associate-tag-option-with-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id tag-p3abc2pkpz5qc
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AssociateTagOptionWithResource](#) na Referência de AWS CLI Comandos.

copy-product

O código de exemplo a seguir mostra como usar `copy-product`.

AWS CLI

Para copiar um produto

O `copy-product` exemplo a seguir faz uma cópia do produto especificado, usando um JSON arquivo para passar parâmetros.

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

Conteúdo de `copy-product-input.json`:

```
{  
  "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
tcabcd3syn2xy",  
  "TargetProductName": "copy-of-myproduct",  
  "CopyOptions": [  
    "CopyTags"  
  ]  
}
```

Saída:

```
{
  "CopyProductToken": "copyproduct-abc5defgjdkdji"
}
```

- Para API obter detalhes, consulte [CopyProduct](#) na Referência de AWS CLI Comandos.

create-portfolio-share

O código de exemplo a seguir mostra como usar `create-portfolio-share`.

AWS CLI

Para compartilhar um portfólio com uma conta

O `create-portfolio-share` exemplo a seguir compartilha o portfólio especificado com a conta especificada.

```
aws servicecatalog create-portfolio-share \
  --portfolio-id port-2s6abcdef5wdh4 \
  --account-id 794123456789
```

Esse comando não produz nenhuma saída.

- Para API obter detalhes, consulte [CreatePortfolioShare](#) na Referência de AWS CLI Comandos.

create-portfolio

O código de exemplo a seguir mostra como usar `create-portfolio`.

AWS CLI

Para criar um portfólio

O `create-portfolio` exemplo a seguir cria um portfólio.

```
aws servicecatalog create-portfolio \
  --provider-name my-provider \
  --display-name my-portfolio
```

Saída:

```
{
  "PortfolioDetail": {
    "ProviderName": "my-provider",
    "DisplayName": "my-portfolio",
    "CreatedTime": 1571337221.555,
    "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-2s6xmplq5wdh4",
    "Id": "port-2s6xmplq5wdh4"
  }
}
```

- Para API obter detalhes, consulte [CreatePortfolio](#) na Referência de AWS CLI Comandos.

create-product

O código de exemplo a seguir mostra como usar create-product.

AWS CLI

Para criar um produto

O create-product exemplo a seguir cria um produto usando um JSON arquivo para passar parâmetros.

```
aws servicecatalog create-product \
  --cli-input-json file://create-product-input.json
```

Conteúdo de 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"
}
}
```

Saída:

```
{
  "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"
    }
  }
}

```

- Para API obter detalhes, consulte [CreateProduct](#) na Referência de AWS CLI Comandos.

create-provisioning-artifact

O código de exemplo a seguir mostra como usar `create-provisioning-artifact`.

AWS CLI

Para criar um artefato de provisionamento

O `create-provisioning-artifact` exemplo a seguir cria um artefato de provisionamento usando um JSON arquivo para transmitir parâmetros.

```

aws servicecatalog create-provisioning-artifact \
  --cli-input-json file://create-provisioning-artifact-input.json

```

Conteúdo de `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"
  }
}

```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [CreateProvisioningArtifact](#) na Referência de AWS CLI Comandos.

create-tag-option

O código de exemplo a seguir mostra como usar `create-tag-option`.

AWS CLI

Para criar um `TagOption`

O `create-tag-option` exemplo a seguir cria um `TagOption`.

```
aws servicecatalog create-tag-option
  --key 1234
  --value name
```

Saída:

```
{
  "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
  }
}
```

```
}  
}
```

- Para API obter detalhes, consulte [CreateTagOption](#) na Referência de AWS CLI Comandos.

delete-portfolio-share

O código de exemplo a seguir mostra como usar delete-portfolio-share.

AWS CLI

Para parar de compartilhar um portfólio com uma conta

O delete-portfolio-share exemplo a seguir interrompe o compartilhamento do portfólio com a conta especificada.

```
aws servicecatalog delete-portfolio-share \  
  --portfolio-id port-2s6abcdq5wdh4 \  
  --account-id 123456789012
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePortfolioShare](#) na Referência de AWS CLI Comandos.

delete-portfolio

O código de exemplo a seguir mostra como usar delete-portfolio.

AWS CLI

Para excluir um portfólio

O delete-portfolio exemplo a seguir exclui o portfólio especificado.

```
aws servicecatalog delete-portfolio \  
  --id port-abcdlx4gox4do
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePortfolio](#) na Referência de AWS CLI Comandos.

delete-product

O código de exemplo a seguir mostra como usar delete-product.

AWS CLI

Para excluir um produto

O delete-product exemplo a seguir exclui o produto especificado.

```
aws servicecatalog delete-product \  
  --id prod-abcdcek6yhbx1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteProduct](#) na Referência de AWS CLI Comandos.

delete-provisioning-artifact

O código de exemplo a seguir mostra como usar delete-provisioning-artifact.

AWS CLI

Para excluir um artefato de provisionamento

O delete-provisioning-artifact exemplo a seguir exclui o artefato de provisionamento especificado.

```
aws servicecatalog delete-provisioning-artifact \  
  --product-id prod-abc2uebup1cpw \  
  --provisioning-artifact-id pa-pqabcddii7ouc
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteProvisioningArtifact](#) na Referência de AWS CLI Comandos.

delete-tag-option

O código de exemplo a seguir mostra como usar delete-tag-option.

AWS CLI

Para excluir um TagOption

O `delete-tag-option` exemplo a seguir exclui o especificado TagOption.

```
aws servicecatalog delete-tag-option \  
  --id tag-iabcdn4fzjjms
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteTagOption](#) na Referência de AWS CLI Comandos.

`describe-copy-product-status`

O código de exemplo a seguir mostra como usar `describe-copy-product-status`.

AWS CLI

Para descrever o status da operação do produto de cópia

O `describe-copy-product-status` exemplo a seguir exibe o status atual da operação do produto de cópia assíncrona especificada.

```
aws servicecatalog describe-copy-product-status \  
  --copy-product-token copyproduct-znn5tf5abcd3w
```

Saída:

```
{  
  "CopyProductStatus": "SUCCEEDED",  
  "TargetProductId": "prod-os6hog7abcdt2"  
}
```

- Para API obter detalhes, consulte [DescribeCopyProductStatus](#) na Referência de AWS CLI Comandos.

`describe-portfolio`

O código de exemplo a seguir mostra como usar `describe-portfolio`.

AWS CLI

Para descrever um portfólio

O `describe-portfolio` exemplo a seguir exibe detalhes do portfólio especificado.

```
aws servicecatalog describe-portfolio \  
  --id port-2s6abcdq5wdh4
```

Saída:

```
{  
  "TagOptions": [],  
  "PortfolioDetail": {  
    "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/  
port-2s6abcdq5wdh4",  
    "Id": "port-2s6wuzyyq5wdh4",  
    "CreatedTime": 1571337221.555,  
    "DisplayName": "my-portfolio",  
    "ProviderName": "my-provider"  
  },  
  "Tags": []  
}
```

- Para API obter detalhes, consulte [DescribePortfolio](#) na Referência de AWS CLI Comandos.

`describe-product-as-admin`

O código de exemplo a seguir mostra como usar `describe-product-as-admin`.

AWS CLI

Para descrever um produto como administrador

O `describe-product-as-admin` exemplo a seguir exibe detalhes do produto especificado usando privilégios de administrador.

```
aws servicecatalog describe-product-as-admin \  
  --id prod-abcdcek6yhbx1
```

Saída:

```
{
  "TagOptions": [],
  "ProductViewDetail": {
    "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-
abcdcek6yhbxix",
    "ProductViewSummary": {
      "SupportEmail": "test@amazon.com",
      "Type": "CLOUD_FORMATION_TEMPLATE",
      "Distributor": "test-distributor",
      "ShortDescription": "test-description",
      "Owner": "test-owner",
      "Id": "prodview-wi3l2j4abc6vc",
      "SupportDescription": "test-support",
      "ProductId": "prod-abcdcek6yhbxix",
      "HasDefaultPath": false,
      "Name": "test-product3",
      "SupportUrl": "https://aws.amazon.com"
    },
    "CreatedTime": 1577136715.0,
    "Status": "CREATED"
  },
  "ProvisioningArtifactSummaries": [
    {
      "CreatedTime": 1577136715.0,
      "Description": "test-version-description",
      "ProvisioningArtifactMetadata": {
        "SourceProvisioningArtifactId": "pa-abcdxkkiv5fcm"
      },
      "Name": "test-version-name-3",
      "Id": "pa-abcdxkkiv5fcm"
    }
  ],
  "Tags": [
    {
      "Value": "iad",
      "Key": "region"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeProductAsAdmin](#) na Referência de AWS CLI Comandos.

describe-provisioned-product

O código de exemplo a seguir mostra como usar `describe-provisioned-product`.

AWS CLI

Para descrever um produto provisionado

O `describe-provisioned-product` exemplo a seguir exibe detalhes do produto provisionado especificado.

```
aws servicecatalog describe-provisioned-product \  
  --id pp-dpom27bm4abcd
```

Saída:

```
{  
  "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": []  
}
```

- Para API obter detalhes, consulte [DescribeProvisionedProduct](#) na Referência de AWS CLI Comandos.

describe-provisioning-artifact

O código de exemplo a seguir mostra como usar `describe-provisioning-artifact`.

AWS CLI

Para descrever um artefato de provisionamento

O `describe-provisioning-artifact` exemplo a seguir exibe detalhes do artefato de provisionamento especificado.

```
aws servicecatalog describe-provisioning-artifact \
  --provisioning-artifact-id pa-pcz347abcdcfm \
  --product-id prod-abcdfz3syn2rg
```

Saída:

```
{
  "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"
}
```

- Para API obter detalhes, consulte [DescribeProvisioningArtifact](#) na Referência de AWS CLI Comandos.

describe-tag-option

O código de exemplo a seguir mostra como usar `describe-tag-option`.

AWS CLI

Para descrever um TagOption

O `describe-tag-option` exemplo a seguir exibe detalhes do especificado TagOption.

```
aws servicecatalog describe-tag-option \  
  --id tag-p3tej2abcd5qc
```

Saída:

```
{  
  "TagOptionDetail": {  
    "Active": true,  
    "Id": "tag-p3tej2abcd5qc",  
    "Value": "value-3",  
    "Key": "1234"  
  }  
}
```

- Para API obter detalhes, consulte [DescribeTagOption](#) na Referência de AWS CLI Comandos.

disassociate-principal-from-portfolio

O código de exemplo a seguir mostra como usar `disassociate-principal-from-portfolio`.

AWS CLI

Para dissociar um diretor de um portfólio

O `disassociate-principal-from-portfolio` exemplo a seguir dissocia o principal especificado do portfólio.

```
aws servicecatalog disassociate-principal-from-portfolio \  
  --portfolio-id port-2s6abcdq5wdh4 \  
  --principal-arn arn:aws:iam::123456789012:group/myendusers
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DisassociatePrincipalFromPortfolio](#) na Referência de AWS CLI Comandos.

disassociate-product-from-portfolio

O código de exemplo a seguir mostra como usar `disassociate-product-from-portfolio`.

AWS CLI

Para desassociar um produto de um portfólio

O `disassociate-product-from-portfolio` exemplo a seguir dissocia o produto especificado do portfólio.

```
aws servicecatalog disassociate-product-from-portfolio \  
  --product-id prod-3p5abcdmu3oyk \  
  --portfolio-id port-2s6abcdq5wdh4
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DisassociateProductFromPortfolio](#) na Referência de AWS CLI Comandos.

`disassociate-tag-option-from-resource`

O código de exemplo a seguir mostra como usar `disassociate-tag-option-from-resource`.

AWS CLI

Para dissociar um TagOption de um recurso

O `disassociate-tag-option-from-resource` exemplo a seguir dissocia o especificado TagOption do recurso.

```
aws servicecatalog disassociate-tag-option-from-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id tag-p3abc2pkpz5qc
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DisassociateTagOptionFromResource](#) na Referência de AWS CLI Comandos.

`list-accepted-portfolio-shares`

O código de exemplo a seguir mostra como usar `list-accepted-portfolio-shares`.

AWS CLI

Para listar ações de portfólio aceitas

O `list-accepted-portfolio-shares` exemplo a seguir lista todos os portfólios para os quais o compartilhamento foi aceito por essa conta, incluindo somente os portfólios padrão do Service Catalog.

```
aws servicecatalog list-accepted-portfolio-shares \
  --portfolio-share-type "AWS_SERVICECATALOG"
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListAcceptedPortfolioShares](#) na Referência de AWS CLI Comandos.

list-portfolio-access

O código de exemplo a seguir mostra como usar `list-portfolio-access`.

AWS CLI

Para listar contas com acesso a um portfólio

O `list-portfolio-access` exemplo a seguir lista as AWS contas que têm acesso ao portfólio especificado.

```
aws servicecatalog list-portfolio-access \  
  --portfolio-id port-2s6abcdq5wdh4
```

Saída:

```
{  
  "AccountIds": [  
    "123456789012"  
  ]  
}
```

- Para API obter detalhes, consulte [ListPortfolioAccess](#) na Referência de AWS CLI Comandos.

list-portfolios-for-product

O código de exemplo a seguir mostra como usar `list-portfolios-for-product`.

AWS CLI

Para listar portfólios associados a um produto

O `list-portfolios-for-product` exemplo a seguir lista os portfólios associados ao produto especificado.

```
aws servicecatalog list-portfolios-for-product \  
  --product-id prod-abcdfz3syn2rg
```

Saída:

```
{  
  "PortfolioDetails": [  
    {  
      "Name": "Portfolio 1"  
    },  
    {  
      "Name": "Portfolio 2"  
    }  
  ]  
}
```

```

    {
      "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"
    }
  ]
}

```

- Para API obter detalhes, consulte [ListPortfoliosForProduct](#) na Referência de AWS CLI Comandos.

list-portfolios

O código de exemplo a seguir mostra como usar `list-portfolios`.

AWS CLI

Para listar portfólios

O `list-portfolios` exemplo a seguir lista os portfólios do Service Catalog na região atual.

```
aws servicecatalog list-portfolios
```

Saída:

```

{
  "PortfolioDetails": [
    {
      "CreatedTime": 1559665256.348,
      "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmlst4ei",

```

```
        "DisplayName": "my-portfolio",
        "Id": "port-5pzcxmlst4ei",
        "ProviderName": "my-user"
    }
]
}
```

- Para API obter detalhes, consulte [ListPortfolios](#) na Referência de AWS CLI Comandos.

list-principals-for-portfolio

O código de exemplo a seguir mostra como usar `list-principals-for-portfolio`.

AWS CLI

Para listar todos os principais de um portfólio

O `list-principals-for-portfolio` exemplo a seguir lista todos os principais do portfólio especificado.

```
aws servicecatalog list-principals-for-portfolio \
  --portfolio-id port-2s6abcdq5wdh4
```

Saída:

```
{
  "Principals": [
    {
      "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
      "PrincipalType": "IAM"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListPrincipalsForPortfolio](#) na Referência de AWS CLI Comandos.

list-provisioning-artifacts

O código de exemplo a seguir mostra como usar `list-provisioning-artifacts`.

AWS CLI

Para listar todos os artefatos de provisionamento de um produto

O `list-provisioning-artifacts` exemplo a seguir lista todos os artefatos de provisionamento do produto especificado.

```
aws servicecatalog list-provisioning-artifacts \  
  --product-id prod-nfi2abcdefgcpw
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListProvisioningArtifacts](#) na Referência de AWS CLI Comandos.

list-resources-for-tag-option

O código de exemplo a seguir mostra como usar `list-resources-for-tag-option`.

AWS CLI

Para listar recursos associados a um TagOption

O `list-resources-for-tag-option` exemplo a seguir lista os recursos associados ao especificado TagOption.

```
aws servicecatalog list-resources-for-tag-option \  
  --tag-option-id tag-p3tej2abcd5qc
```

Saída:

```
{  
  "ResourceDetails": [  
    {  
      "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-  
abcdfz3syn2rg",  
      "Name": "my product",  
      "Description": "description",  
      "CreatedTime": 1562097906.0,  
      "Id": "prod-abcdfz3syn2rg"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListResourcesForTagOption](#) na Referência de AWS CLI Comandos.

list-tag-options

O código de exemplo a seguir mostra como usar `list-tag-options`.

AWS CLI

O `list-tag-options` exemplo a seguir lista todos os valores de TagOptions.

```
aws servicecatalog list-tag-options
```

Saída:

```
{
```

```

    "TagOptionDetails": [
      {
        "Value": "newvalue",
        "Active": true,
        "Id": "tag-iabcdn4fzjjms",
        "Key": "1234"
      },
      {
        "Value": "value1",
        "Active": true,
        "Id": "tag-e3abcdvmwvrzy",
        "Key": "key"
      }
    ]
  }
}

```

- Para API obter detalhes, consulte [ListTagOptions](#) na Referência de AWS CLI Comandos.

provision-product

O código de exemplo a seguir mostra como usar `provision-product`.

AWS CLI

Para provisionar um produto

O `provision-product` exemplo a seguir provisiona o produto especificado usando o artefato de provisionamento especificado.

```

aws servicecatalog provision-product \
  --product-id prod-abcdfz3syn2rg \
  --provisioning-artifact-id pa-abc347pcscfm \
  --provisioned-product-name "mytestppname3"

```

Saída:

```

{
  "RecordDetail": {
    "RecordId": "rec-tfuawdabcdege",
    "CreatedTime": 1577222793.362,
    "ProvisionedProductId": "pp-abcd27bm4mldq",
    "PathId": "lpv2-abcdg3jp6t5k6",

```

```
"RecordErrors": [],
"ProductId": "prod-abcdefz3syn2rg",
"UpdatedTime": 1577222793.362,
"RecordType": "PROVISION_PRODUCT",
"ProvisionedProductName": "mytestppname3",
"ProvisioningArtifactId": "pa-pcz347abcdcfm",
"RecordTags": [],
"Status": "CREATED",
"ProvisionedProductType": "CFN_STACK"
}
}
```

- Para API obter detalhes, consulte [ProvisionProduct](#) na Referência de AWS CLI Comandos.

reject-portfolio-share

O código de exemplo a seguir mostra como usar `reject-portfolio-share`.

AWS CLI

Para rejeitar um compartilhamento do portfólio

O `reject-portfolio-share` exemplo a seguir rejeita a participação do portfólio em questão.

```
aws servicecatalog reject-portfolio-share \
  --portfolio-id port-2s6wuabcdefghijk
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RejectPortfolioShare](#) na Referência de AWS CLI Comandos.

scan-provisioned-products

O código de exemplo a seguir mostra como usar `scan-provisioned-products`.

AWS CLI

Para listar todos os produtos provisionados disponíveis

O `scan-provisioned-products` exemplo a seguir lista os produtos provisionados disponíveis.

```
aws servicecatalog scan-provisioned-products
```

Saída:

```
{
  "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:
Validation error; 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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ScanProvisionedProducts](#) na Referência de AWS CLI Comandos.

search-products-as-admin

O código de exemplo a seguir mostra como usar `search-products-as-admin`.

AWS CLI

Para pesquisar produtos com privilégios de administrador

O `search-products-as-admin` exemplo a seguir pesquisa produtos com privilégios de administrador, usando uma ID de portfólio como filtro.

```
aws servicecatalog search-products-as-admin \
  --portfolio-id port-5abcd3e5st4ei
```

Saída:

```
{
```

```

"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"
  }
]
}

```

- Para API obter detalhes, consulte [SearchProductsAsAdmin](#) na Referência de AWS CLI Comandos.

search-provisioned-products

O código de exemplo a seguir mostra como usar search-provisioned-products.

AWS CLI

Para pesquisar produtos provisionados

O search-provisioned-products exemplo a seguir pesquisa produtos provisionados que correspondam ao ID do produto especificado, usando um JSON arquivo para transmitir parâmetros.

```

aws servicecatalog search-provisioned-products \
  --cli-input-json file://search-provisioned-products-input.json

```

Conteúdo de search-provisioned-products-input.json:

```

{
  "Filters": {
    "SearchQuery": [

```

```

    "prod-tcjevz3syn2rg"
  ]
}

```

Saída:

```

{
  "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
}

```

- Para API obter detalhes, consulte [SearchProvisionedProducts](#) na Referência de AWS CLI Comandos.

update-portfolio

O código de exemplo a seguir mostra como usar update-portfolio.

AWS CLI

Para atualizar um portfólio

O update-portfolio exemplo a seguir atualiza o nome do portfólio especificado.

```

aws servicecatalog update-portfolio \
  --id port-5abcd3e5st4ei \
  --display-name "New portfolio name"

```

Saída:

```

{
  "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": []
}
```

- Para API obter detalhes, consulte [UpdatePortfolio](#) na Referência de AWS CLI Comandos.

update-product

O código de exemplo a seguir mostra como usar update-product.

AWS CLI

Para atualizar um produto

O update-product exemplo a seguir atualiza o nome e o proprietário do produto especificado.

```
aws servicecatalog update-product \
  --id prod-os6abc7drqlt2 \
  --name "New product name" \
  --owner "Updated product owner"
```

Saída:

```
{
  "Tags": [
    {
      "Value": "iad",
      "Key": "region"
    }
  ],
  "ProductViewDetail": {
    "ProductViewSummary": {
      "Owner": "Updated product owner",
      "ProductId": "prod-os6abc7drqlt2",
      "Distributor": "test-distributor",
      "SupportUrl": "https://aws.amazon.com",
      "Name": "New product name",
      "ShortDescription": "test-description",
      "HasDefaultPath": false,
      "Id": "prodview-6abcdgrfhvidy",
```

```

        "SupportDescription": "test-support",
        "SupportEmail": "test@amazon.com",
        "Type": "CLOUD_FORMATION_TEMPLATE"
    },
    "Status": "CREATED",
    "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
os6abc7drqlt2",
    "CreatedTime": 1577136255.0
}
}

```

- Para API obter detalhes, consulte [UpdateProduct](#) na Referência de AWS CLI Comandos.

update-provisioning-artifact

O código de exemplo a seguir mostra como usar `update-provisioning-artifact`.

AWS CLI

Para atualizar um artefato de provisionamento

O `update-provisioning-artifact` exemplo a seguir atualiza o nome e a descrição do artefato de provisionamento especificado, usando um JSON arquivo para transmitir parâmetros.

```

aws servicecatalog update-provisioning-artifact \
  --cli-input-json file://update-provisioning-artifact-input.json

```

Conteúdo de `update-provisioning-artifact-input.json`:

```

{
  "ProductId": "prod-abcdefz3syn2rg",
  "ProvisioningArtifactId": "pa-pcz347abcdcfm",
  "Name": "updated name",
  "Description": "updated description"
}

```

Saída:

```

{
  "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
    }
}

```

- Para API obter detalhes, consulte [UpdateProvisioningArtifact](#) na Referência de AWS CLI Comandos.

update-tag-option

O código de exemplo a seguir mostra como usar `update-tag-option`.

AWS CLI

Para atualizar um TagOption

O `update-tag-option` exemplo a seguir atualiza o valor de aTagOption, usando o JSON arquivo especificado.

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-input.json
```

Conteúdo de `update-tag-option-input.json`:

```

{
  "Id": "tag-iabcdn4fzjjms",
  "Value": "newvalue",
  "Active": true
}

```

Saída:

```

{
  "TagOptionDetail": {
    "Value": "newvalue",

```

```
    "Key": "1234",
    "Active": true,
    "Id": "tag-iabcdn4fzjjms"
  }
}
```

- Para API obter detalhes, consulte [UpdateTagOption](#) na Referência de AWS CLI Comandos.

Exemplos de Service Quotas usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com Service Quotas.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-aws-default-service-quota

O código de exemplo a seguir mostra como usar `get-aws-default-service-quota`.

AWS CLI

Para descrever uma cota de serviço padrão

O `get-aws-default-service-quota` exemplo a seguir exibe detalhes da cota especificada.

```
aws service-quotas get-aws-default-service-quota \
  --service-code ec2 \
  --quota-code L-1216C47A
```

Saída:

```
{
  "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"
    }
  }
}
```

- Para API obter detalhes, consulte [GetAwsDefaultServiceQuota](#) na Referência de AWS CLI Comandos.

get-requested-service-quota-change

O código de exemplo a seguir mostra como usar `get-requested-service-quota-change`.

AWS CLI

Para descrever uma solicitação de aumento de cota de serviço

O `get-requested-service-quota-change` exemplo a seguir descreve a solicitação de aumento de cota especificada.

```
aws service-quotas get-requested-service-quota-change \
  --request-id d187537d15254312a9609aa51bbf7624u7W49tP0
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [GetRequestedServiceQuotaChange](#) na Referência de AWS CLI Comandos.

get-service-quota

O código de exemplo a seguir mostra como usar `get-service-quota`.

AWS CLI

Para descrever uma cota de serviço

O `get-service-quota` exemplo a seguir exibe detalhes sobre a cota especificada.

```
aws service-quotas get-service-quota \
  --service-code ec2 \
  --quota-code L-1216C47A
```

Saída:

```
{
```

```

    "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"
      }
    }
  }
}

```

- Para API obter detalhes, consulte [GetServiceQuota](#) na Referência de AWS CLI Comandos.

list-aws-default-service-quotas

O código de exemplo a seguir mostra como usar `list-aws-default-service-quotas`.

AWS CLI

Para listar as cotas padrão de um serviço

O `list-aws-default-service-quotas` exemplo a seguir lista os valores padrão das cotas do serviço especificado.

```
aws service-quotas list-aws-default-service-quotas \
  --service-code xray
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [ListAwsDefaultServiceQuotas](#) na Referência de AWS CLI Comandos.

list-requested-service-quota-change-history-by-quota

O código de exemplo a seguir mostra como usar `list-requested-service-quota-change-history-by-quota`.

AWS CLI

Para listar suas solicitações de aumento de cota

O `list-requested-service-quota-change-history-by-quota` exemplo a seguir lista as solicitações de aumento de cota para a cota especificada.

```
aws service-quotas list-requested-service-quota-change-history-by-quota \
  --service-code ec2 \
  --quota-code L-20F13EBD
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListRequestedServiceQuotaChangeHistoryByQuota](#) Referência de AWS CLI Comandos.

list-requested-service-quota-change-history

O código de exemplo a seguir mostra como usar `list-requested-service-quota-change-history`.

AWS CLI

Para listar suas solicitações de aumento de cota

O `list-requested-service-quota-change-history` exemplo a seguir lista as solicitações de aumento de cota para o serviço especificado.

```
aws service-quotas list-requested-service-quota-change-history \
  --service-code ec2
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListRequestedServiceQuotaChangeHistory](#) na Referência de AWS CLI Comandos.

list-service-quotas

O código de exemplo a seguir mostra como usar `list-service-quotas`.

AWS CLI

Para listar as cotas de um serviço

O `list-service-quotas` exemplo a seguir exibe detalhes sobre as cotas para AWS CloudFormation.

```
aws service-quotas list-service-quotas \
  --service-code cloudformation
```

Saída:

```
{
  "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
    }
  ]
}
```

```
}
```

- Para API obter detalhes, consulte [ListServiceQuotas](#) na Referência de AWS CLI Comandos.

list-services

O código de exemplo a seguir mostra como usar `list-services`.

AWS CLI

Para listar os serviços disponíveis

O comando a seguir lista os serviços que estão disponíveis nas Cotas de Serviço.

```
aws service-quotas list-services
```

Saída:

```
{
  "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"
    }
  ]
}
```

Você pode adicionar o `--query` parâmetro para filtrar a exibição às informações de seu interesse. O exemplo a seguir exibe somente os códigos de serviço.

```
aws service-quotas list-services \  
  --query Services[*].ServiceCode
```

Saída:

```
[  
  "AWSCloudMap",  
  "access-analyzer",  
  "acm",  
  "acm-pca",  
  "amplify",  
  "apigateway",  
  "application-autoscaling",  
  ...truncated...  
  "xray"  
]
```

- Para API obter detalhes, consulte [ListServices](#) na Referência de AWS CLI Comandos.

request-service-quota-increase

O código de exemplo a seguir mostra como usar `request-service-quota-increase`.

AWS CLI

Para solicitar um aumento da cota de serviço

O `request-service-quota-increase` exemplo a seguir solicita um aumento na cota de serviço especificada.

```
aws service-quotas request-service-quota-increase \  
  --service-code ec2 \  
  --quota-code L-20F13EBD \  
  --desired-value 2
```

Saída:

```
{
```

```
"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"
}
```

- Para API obter detalhes, consulte [RequestServiceQuotaIncrease](#) na Referência de AWS CLI Comandos.

SESExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonSES.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

delete-identity

O código de exemplo a seguir mostra como usar `delete-identity`.

AWS CLI

Para excluir uma identidade

O exemplo a seguir usa o `delete-identity` comando para excluir uma identidade da lista de identidades verificadas com a AmazonSES:

```
aws ses delete-identity --identity user@example.com
```

Para obter mais informações sobre identidades verificadas, consulte Verificação de endereços de e-mail e domínios na Amazon no Guia SES do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [DeletIdentity](#) na Referência de AWS CLI Comandos.

get-identity-dkim-attributes

O código de exemplo a seguir mostra como usar `get-identity-dkim-attributes`.

AWS CLI

Para obter os DKIM atributos do Amazon SES Easy para uma lista de identidades

O exemplo a seguir usa o `get-identity-dkim-attributes` comando para recuperar os DKIM atributos do Amazon SES Easy para uma lista de identidades:

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

Saída:

```
{
  "DkimAttributes": {
    "example.com": {
      "DkimTokens": [
        "EXAMPLEejcs5xoyqytjsotsijas7236gr",
        "EXAMPLEjr76cvoc6mysspnioorxsn6ep",
        "EXAMPLEkbnkqkhlm2lyz77ppkulerm4k"
      ],
      "DkimEnabled": true,
      "DkimVerificationStatus": "Success"
    },
    "user@example.com": {
```

```
        "DkimEnabled": false,  
        "DkimVerificationStatus": "NotStarted"  
    }  
}  
}
```

Se você chamar esse comando com uma identidade que nunca foi enviada para verificação, essa identidade não aparecerá na saída.

Para obter mais informações sobre o EasyDKIM, consulte Easy DKIM na Amazon SES no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [GetIdentityDkimAttributes](#) na Referência de AWS CLI Comandos.

get-identity-notification-attributes

O código de exemplo a seguir mostra como usar `get-identity-notification-attributes`.

AWS CLI

Para obter os atributos de SES notificação da Amazon para uma lista de identidades

O exemplo a seguir usa o `get-identity-notification-attributes` comando para recuperar os atributos de SES notificação da Amazon para uma lista de identidades:

```
aws ses get-identity-notification-attributes --  
identities "user1@example.com" "user2@example.com"
```

Saída:

```
{  
  "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  
    }  
  }  
}
```



```
    }  
  }  
}
```

Esse comando retorna o status do encaminhamento de feedback por e-mail e, se aplicável, os Amazon Resource Names (ARNs) dos SNS tópicos da Amazon para os quais as notificações de devolução, reclamação e entrega são enviadas.

Se você chamar esse comando com uma identidade que nunca foi enviada para verificação, essa identidade não aparecerá na saída.

Para obter mais informações sobre notificações, consulte [Usando notificações com a Amazon SES](#) no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [GetIdentityNotificationAttributes](#) na Referência de AWS CLI Comandos.

get-identity-verification-attributes

O código de exemplo a seguir mostra como usar `get-identity-verification-attributes`.

AWS CLI

Para obter o status SES de verificação da Amazon para uma lista de identidades

O exemplo a seguir usa o `get-identity-verification-attributes` comando para recuperar o status de SES verificação da Amazon para uma lista de identidades:

```
aws ses get-identity-verification-attributes --  
identities "user1@example.com" "user2@example.com"
```

Saída:

```
{  
  "VerificationAttributes": {  
    "user1@example.com": {  
      "VerificationStatus": "Success"  
    },  
    "user2@example.com": {  
      "VerificationStatus": "Pending"  
    }  
  }  
}
```

```
}  
}
```

Se você chamar esse comando com uma identidade que nunca foi enviada para verificação, essa identidade não aparecerá na saída.

Para obter mais informações sobre identidades verificadas, consulte Verificação de endereços de e-mail e domínios na Amazon no Guia SES do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [GetIdentityVerificationAttributes](#) na Referência de AWS CLI Comandos.

get-send-quota

O código de exemplo a seguir mostra como usar `get-send-quota`.

AWS CLI

Para obter seus limites de SES envio da Amazon

O exemplo a seguir usa o `get-send-quota` comando para retornar seus limites de SES envio da Amazon:

```
aws ses get-send-quota
```

Saída:

```
{  
  "Max24HourSend": 200.0,  
  "SentLast24Hours": 1.0,  
  "MaxSendRate": 1.0  
}
```

`Max24 HourSend` é sua cota de envio, que é o número máximo de e-mails que você pode enviar em um período de 24 horas. A cota de envio reflete um período de tempo acumulado. Toda vez que você tenta enviar um e-mail, a Amazon SES verifica quantos e-mails você enviou nas últimas 24 horas. Desde que o número total de e-mails que você enviou seja menor que a sua cota, sua solicitação de envio será aceito e seus e-mails serão enviados.

`SentLast24Hours` é o número de e-mails que você enviou nas últimas 24 horas.

MaxSendRate é o número máximo de e-mails que você pode enviar por segundo.

Os limites de envio se baseiam em destinatários, e não em mensagens. Por exemplo, um e-mail com dez destinatários conta como dez em sua cota de envio.

Para obter mais informações, consulte Gerenciando seus limites de SES envio da Amazon no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [GetSendQuota](#) na Referência de AWS CLI Comandos.

get-send-statistics

O código de exemplo a seguir mostra como usar get-send-statistics.

AWS CLI

Para obter suas estatísticas de SES envio da Amazon

O exemplo a seguir usa o get-send-statistics comando para retornar suas estatísticas de SES envio da Amazon

```
aws ses get-send-statistics
```

Saída:

```
{
  "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
    }
  ]
}
```

```
}
```

O resultado é uma lista de pontos de dados, representando as duas últimas semanas da atividade de envio. Cada ponto de dados na lista contém estatísticas para um intervalo de 15 minutos.

Neste exemplo, há apenas dois pontos de dados porque os únicos e-mails que o usuário enviou nas últimas duas semanas caíram em dois intervalos de 15 minutos.

Para obter mais informações, consulte Monitorando suas estatísticas de SES uso da Amazon no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [GetSendStatistics](#) na Referência de AWS CLI Comandos.

list-identities

O código de exemplo a seguir mostra como usar `list-identities`.

AWS CLI

Para listar todas as identidades (endereços de e-mail e domínios) de uma conta específica AWS

O exemplo a seguir usa o `list-identities` comando para listar todas as identidades que foram enviadas para verificação com a AmazonSES:

```
aws ses list-identities
```

Saída:

```
{
  "Identities": [
    "user@example.com",
    "example.com"
  ]
}
```

A lista retornada contém todas as identidades, independentemente do status da verificação (verificada, verificação pendente, falha etc.).

Neste exemplo, endereços de e-mail e domínios são retornados porque não especificamos o parâmetro `identity-type`.

Para obter mais informações sobre verificação, consulte [Verificação de endereços de e-mail e domínios SES na Amazon](#) no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [ListIdentities](#) na Referência de AWS CLI Comandos.

send-email

O código de exemplo a seguir mostra como usar `send-email`.

AWS CLI

Para enviar um e-mail formatado usando a Amazon SES

O exemplo a seguir usa o comando `send-email` para enviar um e-mail formatado:

```
aws ses send-email --from sender@example.com --destination file://destination.json
--message file://message.json
```

Saída:

```
{
  "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"
}
```

O destino e a mensagem são estruturas de JSON dados salvas em arquivos `.json` no diretório atual. Esses arquivos são os seguintes:

`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"
    }
  }
}
```

Substitua os endereços de e-mail do remetente e do destinatário por aqueles que você deseja usar. Observe que o endereço de e-mail do remetente deve ser verificado com a AmazonSES. Até que você tenha acesso de produção à AmazonSES, você também deve verificar o endereço de e-mail de cada destinatário, a menos que o destinatário seja o simulador de SES caixa de correio da Amazon. Para obter mais informações sobre verificação, consulte Verificação de endereços de e-mail e domínios SES na Amazon no Guia do desenvolvedor do Amazon Simple Email Service.

O ID da mensagem na saída indica que a chamada para send-email foi bem-sucedida.

Se você não receber o e-mail, verifique a caixa de lixo eletrônico.

Para obter mais informações sobre o envio de e-mail formatado, consulte Envio de e-mail formatado usando a Amazon SES API no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [SendEmail](#) na Referência de AWS CLI Comandos.

send-raw-email

O código de exemplo a seguir mostra como usar send-raw-email.

AWS CLI

Para enviar um e-mail bruto usando a Amazon SES

O exemplo a seguir usa o `send-raw-email` comando para enviar um e-mail com um TXT anexo:

```
aws ses send-raw-email --raw-message file://message.json
```

Saída:

```
{
  "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

A mensagem bruta é uma estrutura de JSON dados salva em um arquivo nomeado `message.json` no diretório atual. Ele contém o seguinte:

```
{
  "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--"
}
```

Como você pode ver, “Dados” é uma longa sequência de caracteres que contém todo o conteúdo bruto do e-mail em MIME formato, incluindo um anexo chamado `attachment.txt`.

Substitua `sender@example.com` e `recipient@example.com` pelos endereços que você deseja usar. Observe que o endereço de e-mail do remetente deve ser verificado com a AmazonSES. Até que você tenha acesso de produção à AmazonSES, você também deve verificar o endereço de e-mail do destinatário, a menos que o destinatário seja o simulador de SES caixa de correio da Amazon. Para obter mais informações sobre verificação, consulte [Verificação de endereços de e-mail e domínios SES na Amazon](#) no Guia do desenvolvedor do Amazon Simple Email Service.

O ID da mensagem na saída indica que a chamada para `send-raw-email` foi bem-sucedida.

Se você não receber o e-mail, verifique a caixa de lixo eletrônico.

Para obter mais informações sobre o envio de e-mail bruto, consulte [Envio de e-mail bruto usando a Amazon SES API](#) no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [SendRawEmail](#) na Referência de AWS CLI Comandos.

set-identity-dkim-enabled

O código de exemplo a seguir mostra como usar `set-identity-dkim-enabled`.

AWS CLI

Para ativar ou desativar o Easy DKIM para uma identidade SES verificada pela Amazon

O exemplo a seguir usa o `set-identity-dkim-enabled` comando DKIM para desativar um endereço de e-mail verificado:

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Para obter mais informações sobre o EasyDKIM, consulte Easy DKIM na Amazon SES no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [SetIdentityDkimEnabled](#) na Referência de AWS CLI Comandos.

set-identity-feedback-forwarding-enabled

O código de exemplo a seguir mostra como usar `set-identity-feedback-forwarding-enabled`.

AWS CLI

Para ativar ou desativar o encaminhamento de feedback por e-mail de devolução e reclamação para uma identidade verificada pela Amazon SES

O exemplo a seguir usa o `set-identity-feedback-forwarding-enabled` comando para permitir que um endereço de e-mail verificado receba notificações de devolução e reclamação por e-mail:

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --forwarding-enabled
```

Você precisa receber notificações de devolução e reclamação pela Amazon SNS ou pelo encaminhamento de feedback por e-mail. Portanto, você só pode desativar o encaminhamento de feedback por e-mail se selecionar um SNS tópico da Amazon para notificações de devolução e reclamação.

Para obter mais informações sobre notificações, consulte Usando notificações com a Amazon SES no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [SetIdentityFeedbackForwardingEnabled](#) na Referência de AWS CLI Comandos.

set-identity-notification-topic

O código de exemplo a seguir mostra como usar `set-identity-notification-topic`.

AWS CLI

Para definir o SNS tópico da Amazon no qual a Amazon SES publicará notificações de devolução, reclamação e/ou entrega para uma identidade verificada

O exemplo a seguir usa o `set-identity-notification-topic` comando para especificar o SNS tópico da Amazon para o qual um endereço de e-mail verificado receberá notificações de devolução:

```
aws ses set-identity-notification-topic --identity user@example.com --notification-type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

Para obter mais informações sobre notificações, consulte Usando notificações com a Amazon SES no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [SetIdentityNotificationTopic](#) na Referência de AWS CLI Comandos.

verify-domain-dkim

O código de exemplo a seguir mostra como usar `verify-domain-dkim`.

AWS CLI

Para gerar DKIM tokens de um domínio verificado para DKIM assinatura com a Amazon SES

O exemplo a seguir usa o `verify-domain-dkim` comando para gerar DKIM tokens para um domínio que foi verificado com a AmazonSES:

```
aws ses verify-domain-dkim --domain example.com
```

Saída:

```
{
  "DkimTokens": [
    "EXAMPLEq76owjnks31nluwg65scbemvw",
    "EXAMPLEi3dnsj67hstzaj673klariwx2",
    "EXAMPLEwfbtcukvimehexktmdtaz6naj"
  ]
}
```

Para configurar DKIM, você deve usar os DKIM tokens devolvidos para atualizar DNS as configurações do seu domínio com CNAME registros que apontam para chaves DKIM públicas hospedadas pela AmazonSES. Para obter mais informações, consulte Easy DKIM in Amazon SES no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [VerifyDomainDkim](#) na Referência de AWS CLI Comandos.

verify-domain-identity

O código de exemplo a seguir mostra como usar `verify-domain-identity`.

AWS CLI

Para verificar um domínio com a Amazon SES

O exemplo a seguir usa o comando `verify-domain-identity` para verificar um domínio:

```
aws ses verify-domain-identity --domain example.com
```

Saída:

```
{
  "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wmvjuEXAMPLE"
}
```

Para concluir a verificação do domínio, você deve adicionar um TXT registro com o token de verificação devolvido às DNS configurações do seu domínio. Para obter mais informações, consulte Verificação de domínios SES na Amazon no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [VerifyDomainIdentity](#) na Referência de AWS CLI Comandos.

verify-email-identity

O código de exemplo a seguir mostra como usar `verify-email-identity`.

AWS CLI

Para verificar um endereço de e-mail com a Amazon SES

O exemplo a seguir usa o comando `verify-email-identity` para verificar um endereço de e-mail:

```
aws ses verify-email-identity --email-address user@example.com
```

Antes de enviar um e-mail usando a AmazonSES, você deve verificar o endereço ou domínio do qual você está enviando o e-mail para provar que você é o proprietário. Se você ainda não tem acesso à produção, você também precisa verificar todos os endereços de e-mail para os quais você envia e-mails, exceto os endereços de e-mail fornecidos pelo simulador de SES caixa de correio da Amazon.

Depois `verify-email-identity` de ser chamado, o endereço de e-mail receberá um e-mail de verificação. O usuário deve clicar no link do e-mail para concluir o processo de verificação.

Para obter mais informações, consulte [Verificação de endereços de e-mail na Amazon SES](#) no Guia do desenvolvedor do Amazon Simple Email Service.

- Para API obter detalhes, consulte [VerifyEmailIdentity](#) na Referência de AWS CLI Comandos.

Exemplos de Shield usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Shield.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-drt-log-bucket

O código de exemplo a seguir mostra como usar `associate-drt-log-bucket`.

AWS CLI

Para autorizar o a DRT acessar um bucket do Amazon S3

O `associate-drt-log-bucket` exemplo a seguir cria uma associação entre o DRT e o bucket do S3 especificado. Isso permite que DRT o acesse o bucket em nome da conta. :

```
aws shield associate-drt-log-bucket \  
  --log-bucket flow-logs-for-website-lb
```

Este comando não produz saída.

Para obter mais informações, consulte [Autorize a equipe de DDoS resposta](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [AssociateDrtLogBucket](#) na Referência de AWS CLI Comandos.

associate-drt-role

O código de exemplo a seguir mostra como usar `associate-drt-role`.

AWS CLI

Para autorizar o DRT a mitigar possíveis ataques em seu nome

O `associate-drt-role` exemplo a seguir cria uma associação entre a DRT e a função especificada. Eles DRT podem usar a função para acessar e gerenciar a conta.

```
aws shield associate-drt-role \  
  --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

Este comando não produz saída.

Para obter mais informações, consulte [Autorize a equipe de DDoS resposta](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [AssociateDrtRole](#) na Referência de AWS CLI Comandos.

create-protection

O código de exemplo a seguir mostra como usar `create-protection`.

AWS CLI

Para habilitar a proteção AWS Shield Advanced para um único AWS recurso

O `create-protection` exemplo a seguir ativa a proteção Shield Advanced para a AWS CloudFront distribuição especificada.

```
aws shield create-protection \  
  --name "Protection for CloudFront distribution" \  
  --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FX0WY8
```

Saída:

```
{  
  "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Para obter mais informações, consulte [Specify Your Resources to Protect](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [CreateProtection](#) na Referência de AWS CLI Comandos.

create-subscription

O código de exemplo a seguir mostra como usar `create-subscription`.

AWS CLI

Para ativar a proteção AWS Shield Advanced para uma conta

O `create-subscription` exemplo a seguir ativa a proteção Shield Advanced para a conta.

```
aws shield create-subscription
```

Este comando não produz saída.

Para obter mais informações, consulte [Introdução ao AWS Shield Advanced](#) no Guia do desenvolvedor AWS do Shield Advanced.

- Para API obter detalhes, consulte [CreateSubscription](#) na Referência de AWS CLI Comandos.

delete-protection

O código de exemplo a seguir mostra como usar delete-protection.

AWS CLI

Para remover a proteção AWS Shield Advanced de um AWS recurso

O delete-protection exemplo a seguir remove a proteção AWS Shield Advanced especificada.

```
aws shield delete-protection \  
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Este comando não produz saída.

Para obter mais informações, consulte [Removendo o AWS Shield Advanced de um AWS recurso](#) no Guia do desenvolvedor AWS do Shield Advanced.

- Para API obter detalhes, consulte [DeleteProtection](#) na Referência de AWS CLI Comandos.

describe-attack

O código de exemplo a seguir mostra como usar describe-attack.

AWS CLI

Para recuperar uma descrição detalhada de um ataque

O describe-attack exemplo a seguir exibe detalhes sobre o DDoS ataque com o ID de ataque especificado. Você pode obter o ataque IDs executando o list-attacks comando.

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Saída:

```
{
  "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": []
}
```

```
}
```

Para obter mais informações, consulte [Revisando DDoS incidentes](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DescribeAttack](#) na Referência de AWS CLI Comandos.

describe-drt-access

O código de exemplo a seguir mostra como usar `describe-drt-access`.

AWS CLI

Para recuperar uma descrição das autorizações que DRT ele precisa para mitigar ataques em seu nome

O `describe-drt-access` exemplo a seguir recupera a função e as autorizações de bucket do S3 que ele DRT tem, o que permite que ele responda a possíveis ataques em seu nome.

```
aws shield describe-drt-access
```

Saída:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
  "LogBucketList": [
    "flow-logs-for-website-lb"
  ]
}
```

Para obter mais informações, consulte [Autorize a equipe de DDoS resposta](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DescribeDrtAccess](#) na Referência de AWS CLI Comandos.

describe-emergency-contact-settings

O código de exemplo a seguir mostra como usar `describe-emergency-contact-settings`.

AWS CLI

Para recuperar endereços de e-mail de emergência que você tem em arquivo com o DRT

O `describe-emergency-contact-settings` exemplo a seguir recupera os endereços de e-mail que estão arquivados com DRT o da conta. Esses são os endereços que DRT você deve contatar ao responder a um ataque suspeito.

```
aws shield describe-emergency-contact-settings
```

Saída:

```
{
  "EmergencyContactList": [
    {
      "EmailAddress": "ops@example.com"
    },
    {
      "EmailAddress": "ddos-notifications@example.com"
    }
  ]
}
```

Para obter mais informações, consulte [How AWS Shield Works](https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html)< <https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html>> no Shield Advanced Developer Guide.AWS

- Para API obter detalhes, consulte [DescribeEmergencyContactSettings](#) na Referência de AWS CLI Comandos.

describe-protection

O código de exemplo a seguir mostra como usar `describe-protection`.

AWS CLI

Para recuperar os detalhes de uma AWS proteção Shield Advanced

O `describe-protection` exemplo a seguir exibe detalhes sobre a proteção Shield Advanced com o ID especificado. Você pode obter proteção IDs executando o `list-protections` comando.

```
aws shield describe-protection \
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "Protection": {
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Name": "1.2.3.4",
    "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/
eipalloc-0ac1537af40742a6d"
  }
}
```

Para obter mais informações, consulte [Specify Your Resources to Protect](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DescribeProtection](#) na Referência de AWS CLI Comandos.

describe-subscription

O código de exemplo a seguir mostra como usar `describe-subscription`.

AWS CLI

Para recuperar os detalhes da proteção AWS Shield Advanced para a conta

O `describe-subscription` exemplo a seguir exibe detalhes sobre a proteção Shield Advanced fornecida para a conta. :

```
aws shield describe-subscription
```

Saída:

```
{
  "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
    }
]
}
}

```

Para obter mais informações, consulte [Como o AWS Shield funciona](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DescribeSubscription](#) na Referência de AWS CLI Comandos.

disassociate-drt-log-bucket

O código de exemplo a seguir mostra como usar `disassociate-drt-log-bucket`.

AWS CLI

Para remover a autorização DRT para acessar um bucket do Amazon S3 em seu nome

O `disassociate-drt-log-bucket` exemplo a seguir remove a associação entre o DRT e o bucket do S3 especificado. Depois que esse comando for concluído, eles não DRT poderão mais acessar o bucket em nome da conta.

```
aws shield disassociate-drt-log-bucket \
  --log-bucket flow-logs-for-website-lb
```

Este comando não produz saída.

Para obter mais informações, consulte [Autorize a equipe de DDoS resposta](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DisassociateDrtLogBucket](#) na Referência de AWS CLI Comandos.

disassociate-drt-role

O código de exemplo a seguir mostra como usar `disassociate-drt-role`.

AWS CLI

Para remover a autorização DRT para mitigar possíveis ataques em seu nome

O `disassociate-drt-role` exemplo a seguir remove a associação entre a DRT e a conta. Após essa chamada, eles não DRT podem mais acessar ou gerenciar sua conta.

```
aws shield disassociate-drt-role
```

Este comando não produz saída.

Para obter mais informações, consulte [Autorize a equipe de DDoS resposta](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DisassociateDrtRole](#) na Referência de AWS CLI Comandos.

get-subscription-state

O código de exemplo a seguir mostra como usar `get-subscription-state`.

AWS CLI

Para recuperar o estado atual da assinatura AWS Shield Advanced da conta

O `get-subscription-state` exemplo a seguir recupera o estado da proteção Shield Advanced da conta.

```
aws shield get-subscription-state
```

Saída:

```
{
  "SubscriptionState": "ACTIVE"
```



```
}
```

Para obter mais informações, consulte [Como o AWS Shield funciona](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetSubscriptionState](#) na Referência de AWS CLI Comandos.

list-attacks

O código de exemplo a seguir mostra como usar `list-attacks`.

AWS CLI

Para recuperar resumos de ataques do Shield Advanced AWS

O `list-attacks` exemplo a seguir recupera resumos dos ataques para a AWS CloudFront distribuição especificada durante o período especificado. A resposta inclui um ataque IDs que você pode fornecer ao `describe-attack` comando para obter informações detalhadas sobre um ataque.

```
aws shield list-attacks \
  --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXMP22ZVFAOR \
  --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

Saída:

```
{
  "AttackSummaries": [
    {
      "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E1PXMP22ZVFAOR",
      "StartTime": 1529280000.0,
      "EndTime": 1529449200.0,
      "AttackVectors": [
        {
          "VectorType": "SYN_FLOOD"
        }
      ]
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Revisando DDoS incidentes](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListAttacks](#) na Referência de AWS CLI Comandos.

list-protections

O código de exemplo a seguir mostra como usar `list-protections`.

AWS CLI

Para recuperar resumos de proteção do Shield Advanced AWS

O `list-protections` exemplo a seguir recupera resumos das proteções que estão habilitadas para a conta.

```
aws shield list-protections
```

Saída:

```
{
  "Protections": [
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Name": "Protection for CloudFront distribution",
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FX0WY8"
    }
  ]
}
```

Para obter mais informações, consulte [Specify Your Resources to Protect](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListProtections](#) na Referência de AWS CLI Comandos.

update-emergency-contact-settings

O código de exemplo a seguir mostra como usar `update-emergency-contact-settings`.

AWS CLI

Para definir os endereços de e-mail de emergência que estão arquivados com o DRT

O `update-emergency-contact-settings` exemplo a seguir define dois endereços de e-mail que DRT devem ser contatados ao responder a um ataque suspeito.

```
aws shield update-emergency-contact-settings \  
    --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-  
notifications@example.com
```

Este comando não produz saída.

Para obter mais informações, consulte [Como o AWS Shield funciona](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateEmergencyContactSettings](#) na Referência de AWS CLI Comandos.

update-subscription

O código de exemplo a seguir mostra como usar `update-subscription`.

AWS CLI

Para modificar a assinatura AWS Shield Advanced da conta

O `update-subscription` exemplo a seguir permite a renovação automática da assinatura AWS Shield Advanced para a conta.

```
aws shield update-subscription \  
    --auto-renew ENABLED
```

Este comando não produz saída.

Para obter mais informações, consulte [Como o AWS Shield funciona](#) no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateSubscription](#) na Referência de AWS CLI Comandos.

Exemplos de signatários usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Signer.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

cancel-signing-profile

O código de exemplo a seguir mostra como usar `cancel-signing-profile`.

AWS CLI

Para excluir um perfil de assinatura

O `cancel-signing-profile` exemplo a seguir remove um perfil de assinatura existente do AWS Signer.

```
aws signer cancel-signing-profile \  
  --profile-name MyProfile1
```

Este comando não produz saída.

- Para API obter detalhes, consulte [CancelSigningProfile](#) na Referência de AWS CLI Comandos.

describe-signing-job

O código de exemplo a seguir mostra como usar `describe-signing-job`.

AWS CLI

Para exibir detalhes sobre um trabalho de assinatura

O `describe-signing-job` exemplo a seguir exibe detalhes sobre o trabalho de assinatura especificado.

```
aws signer describe-signing-job \  
--job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

Saída:

```
{  
  "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  
}
```

- Para API obter detalhes, consulte [DescribeSigningJob](#) na Referência de AWS CLI Comandos.

get-signing-platform

O código de exemplo a seguir mostra como usar `get-signing-platform`.

AWS CLI

Para exibir detalhes sobre uma plataforma de assinatura

O `get-signing-platform` exemplo a seguir exibe detalhes sobre a plataforma de assinatura especificada.

```
aws signer get-signing-platform \  
--platform-id AmazonFreeRTOS-TI-CC3220SF
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [GetSigningPlatform](#) na Referência de AWS CLI Comandos.

get-signing-profile

O código de exemplo a seguir mostra como usar `get-signing-profile`.

AWS CLI

Para exibir detalhes sobre um perfil de assinatura

O `get-signing-profile` exemplo a seguir exibe detalhes sobre o perfil de assinatura especificado.

```
aws signer get-signing-profile \  
  --profile-name MyProfile3
```

Saída:

```
{  
  "platformId": "AmazonFreeRTOS-TI-CC3220SF",  
  "profileName": "MyProfile3",  
  "status": "Active",  
  "signingMaterial": {  
    "certificateArn": "arn:aws:acm:us-  
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"  
  }  
}
```

- Para API obter detalhes, consulte [GetSigningProfile](#) na Referência de AWS CLI Comandos.

list-signing-jobs

O código de exemplo a seguir mostra como usar `list-signing-jobs`.

AWS CLI

Para listar todos os trabalhos de assinatura

O `list-signing-jobs` exemplo a seguir exibe detalhes sobre todos os trabalhos de assinatura da conta.

aws signer list-signing-jobs

Neste exemplo, dois trabalhos são retornados, um bem-sucedido e outro com falha.

```
{
  "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"
    }
  ]
}
```



```
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListSigningJobs](#) na Referência de AWS CLI Comandos.

list-signing-platforms

O código de exemplo a seguir mostra como usar `list-signing-platforms`.

AWS CLI

Para listar todas as plataformas de assinatura

O `list-signing-platforms` exemplo a seguir exibe detalhes sobre todas as plataformas de assinatura disponíveis.

```
aws signer list-signing-platforms
```

Saída:

```
{  
  "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"
    ]
}
}
]
}

```

- Para API obter detalhes, consulte [ListSigningPlatforms](#) na Referência de AWS CLI Comandos.

list-signing-profiles

O código de exemplo a seguir mostra como usar `list-signing-profiles`.

AWS CLI

Para listar todos os perfis de assinatura

O `list-signing-profiles` exemplo a seguir exibe detalhes sobre todos os perfis de assinatura da conta.

```
aws signer list-signing-profiles
```

Saída:

```
{
```

```

"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"
    }
  }
]
}

```

- Para API obter detalhes, consulte [ListSigningProfiles](#) na Referência de AWS CLI Comandos.

put-signing-profile

O código de exemplo a seguir mostra como usar put-signing-profile.

AWS CLI

Para criar um perfil de assinatura

O put-signing-profile exemplo a seguir cria um perfil de assinatura usando o certificado e a plataforma especificados.

```

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

```

Saída:

```
{
  "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"
}
```

- Para API obter detalhes, consulte [PutSigningProfile](#) na Referência de AWS CLI Comandos.

start-signing-job

O código de exemplo a seguir mostra como usar `start-signing-job`.

AWS CLI

Para começar um trabalho de autógrafos

O `start-signing-job` exemplo a seguir inicia um trabalho de assinatura no código encontrado na fonte especificada. Ele usa o perfil especificado para fazer a assinatura e coloca o código assinado no destino especificado.

```
aws signer start-signing-job \
  --source 's3={bucketName=signer-
source,key=MyCode.rb,version=PNyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4}' \
  --destination 's3={bucketName=signer-destination,prefix=signed-}' \
  --profile-name MyProfile7
```

A saída é a ID do trabalho de assinatura.

```
{
  "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"
}
```

- Para API obter detalhes, consulte [StartSigningJob](#) na Referência de AWS CLI Comandos.

Exemplos de Snowball usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Snowball.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-snowball-usage

O código de exemplo a seguir mostra como usar `get-snowball-usage`.

AWS CLI

Para obter informações sobre o limite do serviço Snowball para sua conta

O `get-snowball-usage` exemplo a seguir exibe informações sobre o limite do serviço Snowball para sua conta e também o número de Snowballs em uso na sua conta.

```
aws snowball get-snowball-usage
```

Saída:

```
{
  "SnowballLimit": 1,
  "SnowballsInUse": 0
}
```

FOR para obter mais informações, consulte [Limites do AWS Snowball Edge no Guia do desenvolvedor do AWS Snowball](#).

- Para API obter detalhes, consulte [GetSnowballUsage](#) na Referência de AWS CLI Comandos.

list-jobs

O código de exemplo a seguir mostra como usar `list-jobs`.

AWS CLI

Para listar os trabalhos atuais do Snowball em sua conta

O `list-jobs` exemplo a seguir exibe uma matriz de `JobListEntry` objetos. Neste exemplo, um único trabalho é listado.

```
aws snowball list-jobs
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Trabalhos para dispositivos AWS Snowball Edge](#) no Guia do desenvolvedor do AWS Snowball.

- Para API obter detalhes, consulte [ListJobs](#) na Referência de AWS CLI Comandos.

SNSExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonSNS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Os cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)
- [Cenários](#)

Ações

add-permission

O código de exemplo a seguir mostra como usar `add-permission`.

AWS CLI

Para adicionar uma permissão a um tópico

O `add-permission` exemplo a seguir adiciona a permissão para 987654321098 que a AWS conta use a `Publish` ação com o tópico especificado em AWS conta123456789012.

```
aws sns add-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission \  
  --aws-account-id 987654321098 \  
  --action-name Publish
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AddPermission](#) na Referência de AWS CLI Comandos.

check-if-phone-number-is-opted-out

O código de exemplo a seguir mostra como usar `check-if-phone-number-is-opted-out`.

AWS CLI

Para verificar a desativação de SMS mensagens de um número de telefone

O `check-if-phone-number-is-opted-out` exemplo a seguir verifica se o número de telefone especificado optou por não receber SMS mensagens da AWS conta atual.

```
aws sns check-if-phone-number-is-opted-out \  
  --phone-number +1555550100
```


Saída:

```
{
  "isOptedOut": false
}
```

- Para API obter detalhes, consulte [CheckIfPhoneNumberIsOptedOut](#) na Referência de AWS CLI Comandos.

confirm-subscription

O código de exemplo a seguir mostra como usar `confirm-subscription`.

AWS CLI

Para confirmar uma assinatura

O `confirm-subscription` comando a seguir conclui o processo de confirmação iniciado quando você se inscreveu em um SNS tópico chamado `my-topic`. O parâmetro `--token` vem da mensagem de confirmação enviada ao endpoint de notificação especificado na chamada de assinatura.

```
aws sns confirm-subscription \
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
  --
token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5a
```

Saída:

```
{
  "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
}
```

- Para API obter detalhes, consulte [ConfirmSubscription](#) na Referência de AWS CLI Comandos.

create-platform-application

O código de exemplo a seguir mostra como usar `create-platform-application`.

AWS CLI

Para criar um aplicativo de plataforma

O `create-platform-application` exemplo a seguir cria um aplicativo da plataforma Google Firebase usando a credencial de plataforma especificada.

```
aws sns create-platform-application \  
  --name MyApplication \  
  --platform GCM \  
  --attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

Saída:

```
{  
  "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/  
  MyApplication"  
}
```

- Para API obter detalhes, consulte [CreatePlatformApplication](#) na Referência de AWS CLI Comandos.

`create-topic`

O código de exemplo a seguir mostra como usar `create-topic`.

AWS CLI

Para criar um tópico do SNS

O `create-topic` exemplo a seguir cria um SNS tópico chamado `my-topic`.

```
aws sns create-topic \  
  --name my-topic
```

Saída:

```
{  
  "ResponseMetadata": {  
    "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"  
  },  
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
```

```
}
```

Para obter mais informações, consulte [Usando a interface de linha de AWS comando com a Amazon SQS e a Amazon SNS](#) no Guia do usuário da interface de linha de AWS comando.

- Para API obter detalhes, consulte [CreateTopic](#) na Referência de AWS CLI Comandos.

delete-endpoint

O código de exemplo a seguir mostra como usar delete-endpoint.

AWS CLI

Para excluir um endpoint de aplicativo da plataforma

O delete-endpoint exemplo a seguir exclui o endpoint do aplicativo de plataforma especificado.

```
aws sns delete-endpoint \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteEndpoint](#) na Referência de AWS CLI Comandos.

delete-platform-application

O código de exemplo a seguir mostra como usar delete-platform-application.

AWS CLI

Para excluir um aplicativo da plataforma

O delete-platform-application exemplo a seguir exclui o aplicativo de plataforma especificado.

```
aws sns delete-platform-application \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/  
MyApplication
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeletePlatformApplication](#) na Referência de AWS CLI Comandos.

delete-topic

O código de exemplo a seguir mostra como usar delete-topic.

AWS CLI

Para excluir um SNS tópico

O delete-topic exemplo a seguir exclui o SNS tópico especificado.

```
aws sns delete-topic \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteTopic](#) na Referência de AWS CLI Comandos.

get-endpoint-attributes

O código de exemplo a seguir mostra como usar get-endpoint-attributes.

AWS CLI

Para listar os atributos do endpoint do aplicativo da plataforma

O get-endpoint-attributes exemplo a seguir lista os atributos do endpoint do aplicativo de plataforma especificado.

```
aws sns get-endpoint-attributes \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/  
  MyApplication/12345678-abcd-9012-efgh-345678901234
```

Saída:

```
{  
  "Attributes": {  
    "Enabled": "true",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

```
}
```

- Para API obter detalhes, consulte [GetEndpointAttributes](#) na Referência de AWS CLI Comandos.

get-platform-application-attributes

O código de exemplo a seguir mostra como usar `get-platform-application-attributes`.

AWS CLI

Para listar os atributos do aplicativo da plataforma

O `get-platform-application-attributes` exemplo a seguir lista os atributos do aplicativo de plataforma especificado.

```
aws sns get-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/  
  MyApplication
```

Saída:

```
{  
  "Attributes": {  
    "Enabled": "true",  
    "SuccessFeedbackSampleRate": "100"  
  }  
}
```

- Para API obter detalhes, consulte [GetPlatformApplicationAttributes](#) na Referência de AWS CLI Comandos.

get-sms-attributes

O código de exemplo a seguir mostra como usar `get-sms-attributes`.

AWS CLI

Para listar os atributos padrão da SMS mensagem

O `get-sms-attributes` exemplo a seguir lista os atributos padrão para o envio de SMS mensagens.

```
aws sns get-sms-attributes
```

Saída:

```
{
  "attributes": {
    "DefaultSenderId": "MyName"
  }
}
```

- Para API obter detalhes, consulte [GetSMSAttributes](#) na Referência de AWS CLI Comandos.

get-subscription-attributes

O código de exemplo a seguir mostra como usar `get-subscription-attributes`.

AWS CLI

Para recuperar atributos de assinatura de um tópico

A seguir, são `get-subscription-attributes` exibidos os atributos da assinatura especificada. Você pode obter o `subscription-arn` partir da saída do `list-subscriptions` comando.

```
aws sns get-subscription-attributes \
  --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-
  topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

Saída:

```
{
  "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"
  }
}
```

```
}

```

- Para API obter detalhes, consulte [GetSubscriptionAttributes](#) na Referência de AWS CLI Comandos.

get-topic-attributes

O código de exemplo a seguir mostra como usar `get-topic-attributes`.

AWS CLI

Para recuperar os atributos de um tópico

O exemplo `get-topic-attributes` a seguir exibe os atributos do tópico especificado.

```
aws sns get-topic-attributes \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Saída:

```
{
  "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"
  }
}
```

- Para API obter detalhes, consulte [GetTopicAttributes](#) na Referência de AWS CLI Comandos.

list-endpoints-by-platform-application

O código de exemplo a seguir mostra como usar `list-endpoints-by-platform-application`.

AWS CLI

Para listar os endpoints de um aplicativo de plataforma

O `list-endpoints-by-platform-application` exemplo a seguir lista os endpoints e os atributos do endpoint para o aplicativo de plataforma especificado.

```
aws sns list-endpoints-by-platform-application \
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication
```

Saída:

```
{
  "Endpoints": [
    {
      "Attributes": {
        "Token": "EXAMPLE12345...",
        "Enabled": "true"
      },
      "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListEndpointsByPlatformApplication](#) na Referência de AWS CLI Comandos.

list-phone-numbers-opted-out

O código de exemplo a seguir mostra como usar `list-phone-numbers-opted-out`.

AWS CLI

Para listar as opções de exclusão de SMS mensagens

O `list-phone-numbers-opted-out` exemplo a seguir lista os números de telefone que optaram por não receber SMS mensagens.

```
aws sns list-phone-numbers-opted-out
```

Saída:

```
{
  "phoneNumbers": [
    "+15555550100"
  ]
}
```

- Para API obter detalhes, consulte [ListPhoneNumbersOptedOut](#) na Referência de AWS CLI Comandos.

list-platform-applications

O código de exemplo a seguir mostra como usar `list-platform-applications`.

AWS CLI

Para listar os aplicativos da plataforma

O `list-platform-applications` exemplo a seguir lista os aplicativos da plataforma para ADM MPNS e.

```
aws sns list-platform-applications
```

Saída:

```
{
  "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"
  }
}
]
```

- Para API obter detalhes, consulte [ListPlatformApplications](#) na Referência de AWS CLI Comandos.

list-subscriptions-by-topic

O código de exemplo a seguir mostra como usar `list-subscriptions-by-topic`.

AWS CLI

Para listar as assinaturas associadas a um tópico

O texto a seguir `list-subscriptions-by-topic` recupera uma lista de SNS assinaturas associadas ao tópico especificado.

```
aws sns list-subscriptions-by-topic \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListSubscriptionsByTopic](#) na Referência de AWS CLI Comandos.

list-subscriptions

O código de exemplo a seguir mostra como usar `list-subscriptions`.

AWS CLI

Para listar suas SNS assinaturas

O `list-subscriptions` exemplo a seguir exibe uma lista das SNS assinaturas em sua AWS conta.

```
aws sns list-subscriptions
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListSubscriptions](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um tópico

O `list-tags-for-resource` exemplo a seguir lista as tags para o SNS tópico específico da Amazon.

```
aws sns list-tags-for-resource \
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic
```

Saída:

```
{
  "Tags": [
    {
      "Key": "Team",
      "Value": "Alpha"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-topics

O código de exemplo a seguir mostra como usar `list-topics`.

AWS CLI

Para listar seus SNS tópicos

O `list-topics` exemplo a seguir lista todos os SNS tópicos da sua AWS conta.

```
aws sns list-topics
```

Saída:

```
{
  "Topics": [
    {
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListTopics](#) na Referência de AWS CLI Comandos.

opt-in-phone-number

O código de exemplo a seguir mostra como usar `opt-in-phone-number`.

AWS CLI

Para optar por receber mensagens SMS

O `opt-in-phone-number` exemplo a seguir permite que o número de telefone especificado receba SMS mensagens.

```
aws sns opt-in-phone-number \  
  --phone-number +15555550100
```

Este comando não produz saída.

- Para API obter detalhes, consulte [OptInPhoneNumber](#) na Referência de AWS CLI Comandos.

publish

O código de exemplo a seguir mostra como usar `publish`.

AWS CLI

Exemplo 1: Para publicar uma mensagem em um tópico:

O `publish` exemplo a seguir publica a mensagem especificada no SNS tópico especificado. A mensagem é proveniente de um arquivo de texto, o que permite incluir quebras de linha.

```
aws sns publish \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \  
  --message file://message.txt
```

Conteúdo de `message.txt`:

```
Hello World  
Second Line
```

Saída:

```
{
  "MessageId": "123a45b6-7890-12c3-45d6-111122223333"
}
```

Exemplo 2: Para publicar uma SMS mensagem em um número de telefone

O exemplo publish a seguir publica a mensagem Hello world! no número de telefone +1-555-555-0100.

```
aws sns publish \
  --message "Hello world!" \
  --phone-number +1-555-555-0100
```

Saída:

```
{
  "MessageId": "123a45b6-7890-12c3-45d6-333322221111"
}
```

- Para API obter detalhes, consulte [Publicar](#) na Referência de AWS CLI Comandos.

put-data-protection-policy

O código de exemplo a seguir mostra como usar put-data-protection-policy.

AWS CLI

Para definir a política de proteção de dados

Exemplo 1: Para impedir que os editores publiquem mensagens com CreditCardNumber

O put-data-protection-policy exemplo a seguir impede que os editores publiquem mensagens com. 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\": {}}}]}'
```

Este comando não produz saída.

Exemplo 2: Para carregar parâmetros de um arquivo

O seguinte `put-data-protection-policy` carrega parâmetros de um arquivo.

```
aws sns put-data-protection-policy \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --data-protection-policy file://policy.json
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutDataProtectionPolicy](#) na Referência de AWS CLI Comandos.

remove-permission

O código de exemplo a seguir mostra como usar `remove-permission`.

AWS CLI

Para remover uma permissão de um tópico

O `remove-permission` exemplo a seguir remove a permissão `Publish-Permission` do tópico especificado.

```
aws sns remove-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RemovePermission](#) na Referência de AWS CLI Comandos.

set-endpoint-attributes

O código de exemplo a seguir mostra como usar `set-endpoint-attributes`.

AWS CLI

Para definir atributos de endpoint

O `set-endpoint-attributes` exemplo a seguir desativa o endpoint do aplicativo de plataforma especificado.

```
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
```

Saída:

```
{  
  "Attributes": {  
    "Enabled": "false",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

- Para API obter detalhes, consulte [SetEndpointAttributes](#) na Referência de AWS CLI Comandos.

set-platform-application-attributes

O código de exemplo a seguir mostra como usar `set-platform-application-attributes`.

AWS CLI

Para definir os atributos do aplicativo da plataforma

O `set-platform-application-attributes` exemplo a seguir define o `EventDeliveryFailure` atributo do aplicativo ARN de plataforma especificado para o SNS tópico especificado da Amazon.

```
aws sns set-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/  
MyApplication \  
  --attributes EventDeliveryFailure=arn:aws:sns:us-  
west-2:123456789012:AnotherTopic
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetPlatformApplicationAttributes](#) na Referência de AWS CLI Comandos.

set-sms-attributes

O código de exemplo a seguir mostra como usar `set-sms-attributes`.

AWS CLI

Para definir atributos de SMS mensagem

O `set-sms-attributes` exemplo a seguir define a ID padrão do remetente SMS das mensagens para `MyName`.

```
aws sns set-sms-attributes \  
  --attributes DefaultSenderId=MyName
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetSMSAttributes](#) na Referência de AWS CLI Comandos.

set-subscription-attributes

O código de exemplo a seguir mostra como usar `set-subscription-attributes`.

AWS CLI

Para definir atributos de assinatura

O `set-subscription-attributes` exemplo a seguir define o `RawMessageDelivery` atributo para uma SQS assinatura.

```
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
```

Este comando não produz saída.

O `set-subscription-attributes` exemplo a seguir define um `FilterPolicy` atributo para uma SQS assinatura.

```
aws sns set-subscription-attributes \  
  --subscription-arn arn:aws:sns:us-east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \  
  --filter-policy myfilterpolicy
```

```
--attribute-name FilterPolicy \  
--attribute-value "{ \"anyMandatoryKey\": [\"any\", \"of\", \"these\"] }"
```

Este comando não produz saída.

O `set-subscription-attributes` exemplo a seguir remove o `FilterPolicy` atributo de uma SQS assinatura.

```
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 "{}"
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetSubscriptionAttributes](#) na Referência de AWS CLI Comandos.

set-topic-attributes

O código de exemplo a seguir mostra como usar `set-topic-attributes`.

AWS CLI

Para definir um atributo para um tópico

O exemplo `set-topic-attributes` a seguir define o atributo `DisplayName` para o tópico especificado.

```
aws sns set-topic-attributes \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --attribute-name DisplayName \  
  --attribute-value MyTopicDisplayName
```

Este comando não produz saída.

- Para API obter detalhes, consulte [SetTopicAttributes](#) na Referência de AWS CLI Comandos.

subscribe

O código de exemplo a seguir mostra como usar `subscribe`.

AWS CLI

Para inscrever-se em um tópico

O comando `subscribe` a seguir inscreve um endereço de e-mail no tópico especificado.

```
aws sns subscribe \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --protocol email \  
  --notification-endpoint my-email@example.com
```

Saída:

```
{  
  "SubscriptionArn": "pending confirmation"  
}
```

- Para API obter detalhes, consulte [Inscrever-se](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar uma tag a um tópico

O `tag-resource` exemplo a seguir adiciona uma tag de metadados ao SNS tópico especificado da Amazon.

```
aws sns tag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tags Key=Team,Value=Alpha
```

Este comando não produz saída.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

unsubscribe

O código de exemplo a seguir mostra como usar `unsubscribe`.

AWS CLI

Para cancelar a assinatura de um tópico

O exemplo `unsubscribe` a seguir exclui a assinatura especificada de um tópico.

```
aws sns unsubscribe \  
  --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

Este comando não produz saída.

- Para API obter detalhes, consulte [Cancelar inscrição](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover uma tag de um tópico

O `untag-resource` exemplo a seguir remove todas as tags com as chaves especificadas do SNS tópico especificado da Amazon.

```
aws sns untag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tag-keys Team
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

Cenários

Criar um endpoint de plataforma para notificações por push

O exemplo de código a seguir mostra como criar um endpoint de plataforma para notificações SNS push da Amazon.

AWS CLI

Para criar um endpoint de aplicação de plataforma

O exemplo `create-platform-endpoint` a seguir cria um endpoint para a aplicação de plataforma especificada usando o token especificado.

```
aws sns create-platform-endpoint \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/  
MyApplication \  
  --token EXAMPLE12345...
```

Saída:

```
{  
  "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234"  
}
```

SQSExemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonSQS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-permission

O código de exemplo a seguir mostra como usar `add-permission`.

AWS CLI

Para adicionar uma permissão a uma fila

Este exemplo permite que a AWS conta especificada envie mensagens para a fila especificada.

Comando:

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue --aws-account-ids 12345EXAMPLE --actions SendMessage
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [AddPermission](#) na Referência de AWS CLI Comandos.

cancel-message-move-task

O código de exemplo a seguir mostra como usar `cancel-message-move-task`.

AWS CLI

Para cancelar uma tarefa de movimentação de mensagem

O `cancel-message-move-task` exemplo a seguir cancela a tarefa de movimentação de mensagens especificada.

```
aws sqs cancel-message-move-task \  
  --task-handle AQEB6nR4...Hz1vZQ==
```

Saída:

```
{  
  "ApproximateNumberOfMessagesMoved": 102  
}
```

Para obter mais informações, consulte [SQSAPIPermissões da Amazon: ações e referência de recursos](#) no Guia do desenvolvedor.

- Para API obter detalhes, consulte [CancelMessageMoveTask](#) na Referência de AWS CLI Comandos.

change-message-visibility-batch

O código de exemplo a seguir mostra como usar `change-message-visibility-batch`.

AWS CLI

Para alterar as visibilidades de tempo limite de várias mensagens como um lote

Este exemplo altera as visibilidades de tempo limite das 2 mensagens especificadas para 10 horas (10 horas * 60 minutos * 60 segundos).

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
```

Arquivo de entrada (`change-message-visibility-batch.json`):

```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEBhz2q...Jf3kaw==",
    "VisibilityTimeout": 36000
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBkTUH...HifSnw==",
    "VisibilityTimeout": 36000
  }
]
```

Saída:

```
{
  "Successful": [
    {
      "Id": "SecondMessage"
    },
    {
      "Id": "FirstMessage"
    }
  ]
}
```

```
]
}
```

- Para API obter detalhes, consulte [ChangeMessageVisibilityBatch](#) na Referência de AWS CLI Comandos.

change-message-visibility

O código de exemplo a seguir mostra como usar `change-message-visibility`.

AWS CLI

Como alterar a visibilidade do tempo limite de uma mensagem

Este exemplo altera a visibilidade do tempo limite da mensagem especificada para 10 horas (10 horas * 60 minutos * 60 segundos).

Comando:

```
aws sqs change-message-visibility --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --visibility-timeout 36000
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [ChangeMessageVisibility](#) na Referência de AWS CLI Comandos.

create-queue

O código de exemplo a seguir mostra como usar `create-queue`.

AWS CLI

Para criar uma fila

Este exemplo cria uma fila com o nome especificado, define o período de retenção da mensagem para 3 dias (3 dias * 24 horas * 60 minutos * 60 segundos) e define a fila de mensagens não entregues da fila especificada com uma contagem máxima de recebimento de 1.000 mensagens.

Comando:

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

Arquivo de entrada (create-queue.json):

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\", \"maxReceiveCount\":\"1000\"}\",
  "MessageRetentionPeriod": "259200"
}
```

Saída:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- Para API obter detalhes, consulte [CreateQueue](#) na Referência de AWS CLI Comandos.

delete-message-batch

O código de exemplo a seguir mostra como usar delete-message-batch.

AWS CLI

Como excluir várias mensagens como um lote

Este exemplo exclui as mensagens especificadas.

Comando:

```
aws sqs delete-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

Arquivo de entrada (delete-message-batch.json):

```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEB1mg1...Z4GuLw=="
  }
]
```

```
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

Saída:

```
{
  "Successful": [
    {
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

- Para API obter detalhes, consulte [DeleteMessageBatch](#) na Referência de AWS CLI Comandos.

delete-message

O código de exemplo a seguir mostra como usar `delete-message`.

AWS CLI

Como excluir uma mensagem

Este exemplo exclui a mensagem especificada.

Comando:

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DeleteMessage](#) na Referência de AWS CLI Comandos.

delete-queue

O código de exemplo a seguir mostra como usar delete-queue.

AWS CLI

Para excluir uma fila

Este exemplo exclui a fila especificada.

Comando:

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [DeleteQueue](#) na Referência de AWS CLI Comandos.

get-queue-attributes

O código de exemplo a seguir mostra como usar get-queue-attributes.

AWS CLI

Como obter os atributos de uma fila

Este exemplo obtém todos os atributos da fila especificada.

Comando:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names ALL
```

Saída:

```
{
  "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"
  }
}

```

Este exemplo obtém somente os atributos especificados de tempo limite de visibilidade e de tamanho máximo da mensagem da fila.

Comando:

```

aws sqs get-queue-attributes --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-
names MaximumMessageSize VisibilityTimeout

```

Saída:

```

{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}

```

- Para API obter detalhes, consulte [GetQueueAttributes](#) na Referência de AWS CLI Comandos.

get-queue-url

O código de exemplo a seguir mostra como usar `get-queue-url`.

AWS CLI

Para obter uma fila URL

Este exemplo obtém as filas especificadas. URL

Comando:

```
aws sqs get-queue-url --queue-name MyQueue
```

Saída:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- Para API obter detalhes, consulte [GetQueueUrl](#) na Referência de AWS CLI Comandos.

list-dead-letter-source-queues

O código de exemplo a seguir mostra como usar `list-dead-letter-source-queues`.

AWS CLI

Para listar filas de origem de letras mortas

Este exemplo lista as filas associadas à fila de origem de letra morta especificada.

Comando:

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

Saída:

```
{
  "queueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- Para API obter detalhes, consulte [ListDeadLetterSourceQueues](#) na Referência de AWS CLI Comandos.

list-message-move-tasks

O código de exemplo a seguir mostra como usar `list-message-move-tasks`.

AWS CLI

Para listar a mensagem, mova as tarefas

O `list-message-move-tasks` exemplo a seguir lista as duas tarefas mais recentes de movimentação de mensagens na fila especificada.

```
aws sqs list-message-move-tasks \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \  
  --max-results 2
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [SQSAPIPermissões da Amazon: ações e referência de recursos](#) no Guia do desenvolvedor.

- Para API obter detalhes, consulte [ListMessageMoveTasks](#) na Referência de AWS CLI Comandos.

list-queue-tags

O código de exemplo a seguir mostra como usar `list-queue-tags`.

AWS CLI

Para listar todas as tags de alocação de custos para uma fila

O `list-queue-tags` exemplo a seguir exibe todas as tags de alocação de custos associadas à fila especificada.

```
aws sqs list-queue-tags \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue
```

Saída:

```
{  
  "Tags": {  
    "Team": "Alpha"  
  }  
}
```

Para obter mais informações, consulte [Listar tags de alocação de custos](#) no Guia do desenvolvedor do Amazon Simple Queue Service.

- Para API obter detalhes, consulte [ListQueueTags](#) na Referência de AWS CLI Comandos.

list-queues

O código de exemplo a seguir mostra como usar `list-queues`.

AWS CLI

Como listar filas

Este exemplo lista todas as filas.

Comando:

```
aws sqs list-queues
```

Saída:

```
{
  "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"
  ]
}
```

Este exemplo lista somente as filas que começam com “My”.

Comando:

```
aws sqs list-queues --queue-name-prefix My
```

Saída:

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- Para API obter detalhes, consulte [ListQueues](#) na Referência de AWS CLI Comandos.

purge-queue

O código de exemplo a seguir mostra como usar `purge-queue`.

AWS CLI

Para limpar uma fila

Este exemplo exclui todas as mensagens na fila especificada.

Comando:

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [PurgeQueue](#) na Referência de AWS CLI Comandos.

receive-message

O código de exemplo a seguir mostra como usar `receive-message`.

AWS CLI

Como receber uma mensagem

Este exemplo recebe até 10 mensagens disponíveis e retorna todos os atributos disponíveis.

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
```

Saída:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIKMSNQ7TEXAMPLE",
        "ApproximateReceiveCount": "5",
```

```

    "SentTimestamp": "1442428276921"
  },
  "MessageAttributes": {
    "PostalCode": {
      "DataType": "String",
      "StringValue": "ABC123"
    },
    "City": {
      "DataType": "String",
      "StringValue": "Any City"
    }
  }
}
]
}

```

Este exemplo recebe a próxima mensagem disponível, retornando somente os `SentTimestamp` atributos `SenderId` e, bem como o atributo da `PostalCode` mensagem.

Comando:

```

aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp --message-attribute-names PostalCode

```

Saída:

```

{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIASZKMSNQ7TEXAMPLE",
        "SentTimestamp": "1442428276921"
      }
    },
    "MessageAttributes": {
      "PostalCode": {
        "DataType": "String",
        "StringValue": "ABC123"
      }
    }
  ]
}

```

```
}  
  }  
} ]  
}
```

- Para API obter detalhes, consulte [ReceiveMessage](#) na Referência de AWS CLI Comandos.

remove-permission

O código de exemplo a seguir mostra como usar `remove-permission`.

AWS CLI

Para remover uma permissão

Este exemplo remove a permissão com o rótulo especificado da fila especificada.

Comando:

```
aws sqs remove-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [RemovePermission](#) na Referência de AWS CLI Comandos.

send-message-batch

O código de exemplo a seguir mostra como usar `send-message-batch`.

AWS CLI

Como enviar várias mensagens como um lote

Este exemplo envia duas mensagens com os corpos da mensagem, os períodos de atraso e os atributos de mensagem especificados para a fila especificada.

Comando:

```
aws sqs send-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

Arquivo de entrada (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"
  }
}
]

```

Saída:

```

{
  "Successful": [
    {
      "MD50fMessageBody": "203c4a38...7943237e",
      "MD50fMessageAttributes": "10809b55...baf283ef",
      "Id": "FuelReport-0001-2015-09-16T140731Z",
      "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
    },
    {
      "MD50fMessageBody": "2cf0159a...c1980595",
      "MD50fMessageAttributes": "55623928...ae354a25",
      "Id": "FuelReport-0002-2015-09-16T140930Z",
      "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
    }
  ]
}

```

- Para API obter detalhes, consulte [SendMessageBatch](#) na Referência de AWS CLI Comandos.

send-message

O código de exemplo a seguir mostra como usar `send-message`.

AWS CLI

Para enviar uma mensagem

Este exemplo envia uma mensagem com o corpo da mensagem, o período de atraso e os atributos da mensagem especificados para a fila especificada.

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
```

Arquivo de entrada (send-message.json):

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
    "DataType": "Binary",
    "BinaryValue": "Hello, World!"
  },
  "Population": {
    "DataType": "Number",
    "StringValue": "1250800"
  }
}
```

Saída:

```
{
  "MD5ofMessageBody": "51b0a325...39163aa0",
  "MD5ofMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```

- Para API obter detalhes, consulte [SendMessage](#) na Referência de AWS CLI Comandos.

set-queue-attributes

O código de exemplo a seguir mostra como usar `set-queue-attributes`.

AWS CLI

Como definir atributos de fila

Este exemplo define a fila especificada com um atraso de entrega de 10 segundos, um tamanho máximo de mensagem de 128 KB (128 KB * 1.024 bytes), um período de retenção de mensagens de 3 dias (3 dias * 24 horas * 60 minutos * 60 segundos), um tempo de espera de recebimento de mensagens de 20 segundos e um tempo limite de visibilidade padrão de 60 segundos. Este exemplo também associa a fila de mensagens não entregues especificada a uma contagem máxima de recebimento de 1.000 mensagens.

Comando:

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

Arquivo de entrada (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"
}
```

Saída:

```
None.
```

- Para API obter detalhes, consulte [SetQueueAttributes](#) na Referência de AWS CLI Comandos.

start-message-move-task

O código de exemplo a seguir mostra como usar `start-message-move-task`.

AWS CLI

Exemplo 1: *Para iniciar uma tarefa de movimentação de mensagem*

O `start-message-move-task` exemplo a seguir inicia uma tarefa de movimentação de mensagens para redirecionar mensagens da fila de mensagens mortas especificada para a fila de origem.

```
aws sqs start-message-move-task \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

Saída:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```

Para obter mais informações, consulte [Este é o título do tópico](#) no Nome do seu guia.

Exemplo 2: *Para iniciar uma tarefa de movimentação de mensagens com uma taxa máxima*

O `start-message-move-task` exemplo a seguir inicia uma tarefa de movimentação de mensagens para redirecionar mensagens da fila de mensagens mortas especificada para a fila de destino especificada a uma taxa máxima de 50 mensagens por segundo.

```
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
```

Saída:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```


Para obter mais informações, consulte [SQSAPIPermissões da Amazon: ações e referência de recursos](#) no Guia do desenvolvedor.

- Para API obter detalhes, consulte [StartMessageMoveTask](#) na Referência de AWS CLI Comandos.

tag-queue

O código de exemplo a seguir mostra como usar tag-queue.

AWS CLI

Para adicionar tags de alocação de custos a uma fila

O tag-queue exemplo a seguir adiciona uma tag de alocação de custos à SQS fila especificada da Amazon.

```
aws sqs tag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tags Priority=Highest
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar tags de alocação de custos](#) no Guia do desenvolvedor do Amazon Simple Queue Service.

- Para API obter detalhes, consulte [TagQueue](#) na Referência de AWS CLI Comandos.

untag-queue

O código de exemplo a seguir mostra como usar untag-queue.

AWS CLI

Para remover as tags de alocação de custos de uma fila

O untag-queue exemplo a seguir remove uma tag de alocação de custos da SQS fila especificada da Amazon.

```
aws sqs untag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tags Priority=Highest
```

```
--tag-keys "Priority"
```

Este comando não produz saída.

Para obter mais informações, consulte [Adicionar tags de alocação de custos](#) no Guia do desenvolvedor do Amazon Simple Queue Service.

- Para API obter detalhes, consulte [UntagQueue](#) na Referência de AWS CLI Comandos.

Exemplos de Storage Gateway usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Storage Gateway.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

describe-gateway-information

O código de exemplo a seguir mostra como usar `describe-gateway-information`.

AWS CLI

Para descrever um gateway

O `describe-gateway-information` comando a seguir retorna metadados sobre o gateway especificado. Para especificar qual gateway descrever, use o Amazon Resource Name (ARN) do gateway no comando.

Este exemplo especifica um gateway com o id `sgw-12A3456B` na conta `123456789012`:

```
aws storagegateway describe-gateway-information --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Esse comando gera um JSON bloco que contém metadados sobre o gateway, como nome, interfaces de rede, fuso horário configurado e estado (se o gateway está em execução ou não).

- Para API obter detalhes, consulte [DescribeGatewayInformation](#) na Referência de AWS CLI Comandos.

list-file-shares

O código de exemplo a seguir mostra como usar `list-file-shares`.

AWS CLI

Para listar compartilhamentos de arquivos

O `command-name` exemplo a seguir lista os widgets disponíveis na sua AWS conta.

```
aws storagegateway list-file-shares \
  --gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [ListFileShares](#) na API Referência do Serviço do AWS Storage Gateway.

- Para API obter detalhes, consulte [ListFileShares](#) na Referência de AWS CLI Comandos.

list-gateways

O código de exemplo a seguir mostra como usar `list-gateways`.

AWS CLI

Para listar gateways para uma conta

O `list-gateways` comando a seguir lista todos os gateways definidos para uma conta:

```
aws storagegateway list-gateways
```

Esse comando gera um JSON bloco que contém uma lista de nomes de recursos da Amazon (ARNs) do gateway.

- Para API obter detalhes, consulte [ListGateways](#) na Referência de AWS CLI Comandos.

list-volumes

O código de exemplo a seguir mostra como usar `list-volumes`.

AWS CLI

Para listar os volumes configurados para um gateway

O `list-volumes` comando a seguir retorna uma lista de volumes configurados para o gateway especificado. Para especificar qual gateway descrever, use o Amazon Resource Name (ARN) do gateway no comando.

Este exemplo especifica um gateway com o id `sgw-12A3456B` na conta `123456789012`:

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Esse comando gera um JSON bloco com uma lista de volumes que inclui o tipo e ARN para cada volume.

- Para API obter detalhes, consulte [ListVolumes](#) na Referência de AWS CLI Comandos.

refresh-cache

O código de exemplo a seguir mostra como usar `refresh-cache`.

AWS CLI

Para atualizar o cache de compartilhamento de arquivos

O `refresh-cache` exemplo a seguir atualiza o cache do compartilhamento de arquivos especificado.

```
aws storagegateway refresh-cache \  
  --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345
```

Saída:

```
{  
  "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345",  
  "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"  
}
```

Para obter mais informações, consulte [ListFileShares](#) na API Referência do Serviço do AWS Storage Gateway.

- Para API obter detalhes, consulte [RefreshCache](#) na Referência de AWS CLI Comandos.

AWS STS exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS STS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

assume-role-with-saml

O código de exemplo a seguir mostra como usar `assume-role-with-saml`.

AWS CLI

Para obter credenciais de curto prazo para uma função autenticada com SAML

O `assume-role-with-saml` comando a seguir recupera um conjunto de credenciais de curto prazo para a IAM função. `TestSam1` A solicitação neste exemplo é autenticada usando a SAML declaração fornecida pelo seu provedor de identidade quando você se autentica nela.

```
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 "VERYLONGENCODEDASSERTIONEXAMPLExzYW1s0kF1ZG1l1bmNlPmJsYW5rPC9zYW1s0kF1ZG1l1bmNlPjwv
+PHNhbw6TmFtZULEIEZvcm1hdD0idXJu0m9hc2lz0m5hbWVz0nRj0lNBTUw6Mi4w0m5hbWVpZC1mb3JtYXQ6dHJhbnN
+PHNhbw6U3ViamVjdENvbmZpcm1hdGlvbiBNZXRob2Q9InVyb3pvczpuYW1lc3p0YzptQU1M0jiuMDpjbTpiZWV"
```

Saída:

```
{
  "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
+Na8KHDcCYRVjJCZEv0AiEA30vJGtw1EcVi0leS2vhs8VdCKFJQPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fP0L9FX9KSYvKTr4rvx3iSI1TJabIQwj2ICCR/oLxBA==",
    "Expiration": "2019-11-01T20:26:47Z"
  },
}
```

```

    "Audience": "https://signin.aws.amazon.com/saml",
    "SubjectType": "transient",
    "PackedPolicySize": "6",
    "NameQualifier": "SbdG0nUkh1i4+EXAMPLExL/jEvs=",
    "Subject": "SamlExample"
  }

```

Para obter mais informações, consulte [Solicitando credenciais de segurança temporárias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [AssumeRoleWithSaml](#) na Referência de AWS CLI Comandos.

assume-role-with-web-identity

O código de exemplo a seguir mostra como usar `assume-role-with-web-identity`.

AWS CLI

Para obter credenciais de curto prazo para uma função autenticada com o Web Identity (OAuth2.0)

O `assume-role-with-web-identity` comando a seguir recupera um conjunto de credenciais de curto prazo para a IAM função. `app1` A solicitação é autenticada com o token de identidade Web fornecido pelo provedor de identidade Web especificado. Duas políticas adicionais são aplicadas à sessão para restringir ainda mais o que o usuário pode fazer. As credenciais retornadas expiram uma hora após serem geradas.

```

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"

```

Saída:

```
{
  "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+FvwqnKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgkBN9bkUDNCJiBeb/AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mRl/+0tkIKG07fAE",
    "Expiration": "2020-05-19T18:06:10+00:00"
  },
  "Provider": "www.amazon.com"
}
```

Para obter mais informações, consulte [Solicitando credenciais de segurança temporárias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [AssumeRoleWithWebIdentity](#) na Referência de AWS CLI Comandos.

assume-role

O código de exemplo a seguir mostra como usar `assume-role`.

AWS CLI

Como assumir um perfil

O `assume-role` comando a seguir recupera um conjunto de credenciais de curto prazo para a IAM função. `s3-access-example`

```
aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
  --role-session-name s3-access-example
```

Saída:


```
{
  "AssumedRoleUser": {
    "AssumedRoleId": "AR0A3XFRBF535PLBIFPI4:s3-access-example",
    "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-example"
  },
  "Credentials": {
    "SecretAccessKey": "9drTJvcXLB89EXAMPLEELB8923FB892xMFI",
    "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEebw/
m3hsj8VBTkP0RGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtY1FVgAUj
+7Indz3LU0aTWk1WKIjHmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFipzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6D19zR0tXoybnlrZIwMLlMi1Kcgo50ytwU=",
    "Expiration": "2016-03-15T00:05:07Z",
    "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
  }
}
```

A saída do comando contém uma chave de acesso, uma chave secreta e um token de sessão que você pode usar para se autenticar na AWS.

Para AWS CLI uso, você pode configurar um perfil nomeado associado a uma função. Quando você usa o perfil, eles AWS CLI chamarão `assume-role` e gerenciarão as credenciais para você. Para obter mais informações, consulte [Usar uma IAM função AWS CLI no](#) Guia do AWS CLI usuário.

- Para API obter detalhes, consulte [AssumeRole](#) na Referência de AWS CLI Comandos.

decode-authorization-message

O código de exemplo a seguir mostra como usar `decode-authorization-message`.

AWS CLI

Para decodificar uma mensagem de autorização codificada retornada em resposta a uma solicitação

O exemplo `decode-authorization-message` a seguir decodifica informações adicionais sobre o status da autorização de uma solicitação de uma mensagem codificada retornada em resposta a uma solicitação da Amazon Web Services.

```
aws sts decode-authorization-message \
  --encoded-message EXAMPLEWodyRNrtLQARDip-
eTA6i6DrLUhHhPQrLWB_LAb15pAKx19mPDLexYcGBreyIKQC1BGBIpBKr3dFDkqwe07e2NMk5j_hmzAiChJN-8oy3Ewi
Ojau7BMjOTWw0tHPHv_Zaz87yENDipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDGvh1gjqZJFUIQoubv78V1RbHNYnK44E
p0u3FZjwYStfvTb3GHs3-6rLribG09jZ0tkkfE6vqx1FzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVPXioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHLI-
tIiNd7tgNTYzDuPQYD2FMDbnp82V9eVmYgtPp5NIeSpuf3f0HanFuBZgENxZQZ2dLH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067P0FrZECLjj0tNPBOZCcuEKEXAM
```

Saída:

```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements\
\":{\n\"items\":[\n\"statementId\":\n\"VisualEditor0\", \"effect\":\n\"DENY\", \"principals\
\":{\n\"items\":[\n\"value\":\n\"AROA123456789EXAMPLE\" ]}], \"principalGroups\
\":{\n\"items\":[\n ]}], \"actions\":{\n\"items\":[\n\"value\":\n\"ec2:RunInstances\
\" ]}], \"resources\":{\n\"items\":[\n\"value\":\n\"*\" ]}], \"conditions\":{\n\"items\
\":[\n ]}], \"failures\":{\n\"items\":[\n ]}], \"context\":{\n\"principal\":{\n\"id\":\
\n\"AROA123456789EXAMPLE:Ana\", \"arn\":\n\"arn:aws:sts:111122223333:assumed-role/\
Developer/Ana\", \"action\":\n\"RunInstances\", \"resource\":\n\"arn:aws:ec2:us-
east-1:111122223333:instance/*\", \"conditions\":{\n\"items\":[\n\"key\":\
\n\"ec2:MetadataHttpPutResponseHopLimit\", \"values\":{\n\"items\":[\n\"value\":\
\n\"2\" ]}], {\n\"key\":\n\"ec2:InstanceMarketType\", \"values\":{\n\"items\":[\n\"value\
\":\n\"on-demand\" ]}], {\n\"key\":\n\"aws:Resource\", \"values\":{\n\"items\":[\n\"value\
\":\n\"instance/*\" ]}], {\n\"key\":\n\"aws:Account\", \"values\":{\n\"items\":[\n\"value\
\":\n\"111122223333\" ]}], {\n\"key\":\n\"ec2:AvailabilityZone\", \"values\":{\n\"items\":\
[\n\"value\":\n\"us-east-1f\" ]}], {\n\"key\":\n\"ec2:ecsOptimized\", \"values\":{\n\"items\
\":[\n\"value\":\n\"false\" ]}], {\n\"key\":\n\"ec2:IsLaunchTemplateResource\", \"values\
\":{\n\"items\":[\n\"value\":\n\"false\" ]}], {\n\"key\":\n\"ec2:InstanceType\", \"values\":\
{\n\"items\":[\n\"value\":\n\"t2.micro\" ]}], {\n\"key\":\n\"ec2:RootDeviceType\", \"values\
\":{\n\"items\":[\n\"value\":\n\"efs\" ]}], {\n\"key\":\n\"aws:Region\", \"values\":{\n\"items\
\":[\n\"value\":\n\"us-east-1\" ]}], {\n\"key\":\n\"ec2:MetadataHttpEndpoint\", \"values\
\":{\n\"items\":[\n\"value\":\n\"enabled\" ]}], {\n\"key\":\n\"aws:Service\", \"values\":\
{\n\"items\":[\n\"value\":\n\"ec2\" ]}], {\n\"key\":\n\"ec2:InstanceID\", \"values\":{\n\"items\
\":[\n\"value\":\n\"*\" ]}], {\n\"key\":\n\"ec2:MetadataHttpTokens\", \"values\":{\n\"items\
\":[\n\"value\":\n\"required\" ]}], {\n\"key\":\n\"aws:Type\", \"values\":{\n\"items\":\
[\n\"value\":\n\"instance\" ]}], {\n\"key\":\n\"ec2:Tenancy\", \"values\":{\n\"items\":\
[\n\"value\":\n\"default\" ]}], {\n\"key\":\n\"ec2:Region\", \"values\":{\n\"items\":[\n\"value\
\":\n\"us-east-1\" ]}], {\n\"key\":\n\"aws:ARN\", \"values\":{\n\"items\":[\n\"value\":\
\n\"arn:aws:ec2:us-east-1:111122223333:instance/*\" ]}}}}}}"}
}
```

Para obter mais informações, consulte [Lógica de avaliação de políticas](#) no Guia AWS IAM do usuário.

- Para API obter detalhes, consulte [DecodeAuthorizationMessage](#) na Referência de AWS CLI Comandos.

get-caller-identity

O código de exemplo a seguir mostra como usar `get-caller-identity`.

AWS CLI

Para obter detalhes sobre a IAM identidade atual

O `get-caller-identity` comando a seguir exibe informações sobre a IAM identidade usada para autenticar a solicitação. O chamador é um IAM usuário.

```
aws sts get-caller-identity
```

Saída:

```
{
  "UserId": "AIDASAMPLEUSERID",
  "Account": "123456789012",
  "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- Para API obter detalhes, consulte [GetCallerIdentity](#) na Referência de AWS CLI Comandos.

get-federation-token

O código de exemplo a seguir mostra como usar `get-federation-token`.

AWS CLI

Para retornar um conjunto de credenciais de segurança temporárias usando as credenciais da chave de acesso IAM do usuário

O exemplo `get-federation-token` a seguir retorna um conjunto de credenciais de segurança temporárias (que consistem em um ID de chave de acesso, uma chave de acesso secreta e um

token de segurança) para um usuário. Você deve chamar a `GetFederationToken` operação usando as credenciais de segurança de longo prazo de um IAM usuário.

```
aws sts get-federation-token \  
  --name Bob \  
  --policy file://myfile.json \  
  --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \  
  --duration-seconds 900
```

Conteúdo de `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": "*"   
    }   
  ]  
}
```

Saída:

```
{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "SessionToken": "EXAMPLEpZ21uX2VjEGoaCXVzLXd1c3Q0tMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42Q0unWMTfKq0DCOP//////////
wEQAxoMNDUy0TI1MTcwNTA3Igxuy3A0puuoLsk3MJwqgQPg8Q0d9HuoClUxq26wnc/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
}
```

Para obter mais informações, consulte [Solicitando credenciais de segurança temporárias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetFederationToken](#) na Referência de AWS CLI Comandos.

get-session-token

O código de exemplo a seguir mostra como usar `get-session-token`.

AWS CLI

Para obter um conjunto de credenciais de curto prazo para uma identidade IAM

O `get-session-token` comando a seguir recupera um conjunto de credenciais de curto prazo para a IAM identidade que está fazendo a chamada. As credenciais resultantes podem ser usadas para solicitações em que a autenticação multifatorial (MFA) é exigida pela política. As credenciais expiram 15 minutos após serem geradas.

```
aws sts get-session-token \
  --duration-seconds 900 \
  --serial-number "YourMFADeviceSerialNumber" \
  --token-code 123456
```

Saída:

```
{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYzEXAMPLEKEY",
    "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwwqNkWRcOIfrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lGkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mRl/+0tkIKG07fAE",
    "Expiration": "2020-05-19T18:06:10+00:00"
  }
}
```

Para obter mais informações, consulte [Solicitando credenciais de segurança temporárias](#) no Guia do AWS IAM usuário.

- Para API obter detalhes, consulte [GetSessionToken](#) na Referência de AWS CLI Comandos.

AWS Support exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS Support.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-attachments-to-set

O código de exemplo a seguir mostra como usar `add-attachments-to-set`.

AWS CLI

Para adicionar um anexo a um conjunto

O `add-attachments-to-set` exemplo a seguir adiciona uma imagem a um conjunto que você pode especificar para um caso de suporte em sua AWS conta.

```
aws support add-attachments-to-set \  
  --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE" \  
  --attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

Saída:

```
{  
  "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE",  
  "expiryTime": "2020-05-14T17:04:40.790+0000"  
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [AddAttachmentsToSet](#) na Referência de AWS CLI Comandos.

add-communication-to-case

O código de exemplo a seguir mostra como usar `add-communication-to-case`.

AWS CLI

Para adicionar comunicação a um caso

O `add-communication-to-case` exemplo a seguir adiciona comunicações a um caso de suporte em sua AWS conta.

```
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-nk5xQlBr70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE"
```

Saída:

```
{
  "result": true
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [AddCommunicationToCasena](#) Referência de AWS CLI Comandos.

create-case

O código de exemplo a seguir mostra como usar create-case.

AWS CLI

Como criar um caso

O create-case exemplo a seguir cria um caso de suporte para sua AWS conta.

```
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"
```

Saída:

```
{
  "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [CreateCasena](#) Referência de AWS CLI Comandos.

describe-attachment

O código de exemplo a seguir mostra como usar describe-attachment.

AWS CLI

Como descrever um anexo

O exemplo de `describe-attachment` a seguir retorna informações sobre o anexo com o ID especificado.

```
aws support describe-attachment \  
  --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-  
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakqLc60-  
iJjL5HqyYGiT1FG8EXAMPLE"
```

Saída:

```
{  
  "attachment": {  
    "fileName": "troubleshoot-screenshot.png",  
    "data": "base64-blob"  
  }  
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [DescribeAttachment](#) na Referência de AWS CLI Comandos.

describe-cases

O código de exemplo a seguir mostra como usar `describe-cases`.

AWS CLI

Como descrever um caso

O `describe-cases` exemplo a seguir retorna informações sobre o caso de suporte especificado em sua AWS conta.

```
aws support describe-cases \  
  --display-id "1234567890" \  
  --after-time "2020-03-23T21:31:47.774Z" \  
  --include-resolved-cases \  
  --language "en" \  
  --no-include-communications \  
  --no-include-communications
```

```
--max-item 1
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [DescribeCases](#) na Referência de AWS CLI Comandos.

describe-communications

O código de exemplo a seguir mostra como usar `describe-communications`.

AWS CLI

Como descrever a comunicação mais recente de um caso

O `describe-communications` exemplo a seguir retorna a comunicação mais recente para o caso de suporte especificado em sua AWS conta.

```
aws support describe-communications \
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
  --after-time "2020-03-23T21:31:47.774Z" \
  --max-item 1
```

Saída:

```
{
  "communications": [
    {
      "body": "I want to learn more about an AWS service.",
      "attachmentSet": [],
      "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47",
      "timeCreated": "2020-05-12T23:12:35.000Z",
      "submittedBy": "Amazon Web Services"
    }
  ],
  "NextToken": "eyJ1ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [DescribeCommunications](#) na Referência de AWS CLI Comandos.

describe-services

O código de exemplo a seguir mostra como usar `describe-services`.

AWS CLI

Para listar AWS serviços e categorias de serviços

O exemplo de `describe-services` a seguir lista as categorias de serviço disponíveis para a solicitação de informações gerais.

```
aws support describe-services \
  --service-code-list general-info
```

Saída:

```
{
  "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"
  }
]
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [DescribeServices](#) na Referência de AWS CLI Comandos.

describe-severity-levels

O código de exemplo a seguir mostra como usar `describe-severity-levels`.

AWS CLI

Como listar os níveis de gravidade disponíveis

O exemplo de `describe-severity-levels` a seguir lista os níveis de gravidade disponíveis para um caso de suporte.

```
aws support describe-severity-levels
```

Saída:

```
{
  "severityLevels": [
    {
      "code": "low",
      "name": "Low"
    },
    {
      "code": "normal",
      "name": "Normal"
    },
    {
      "code": "high",
      "name": "High"
    },
    {
      "code": "urgent",
      "name": "Urgent"
    },
    {
      "code": "critical",
      "name": "Critical"
    }
  ]
}
```

Para obter mais informações, consulte [Choosing a severity](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [DescribeSeverityLevels](#) na Referência de AWS CLI Comandos.

describe-trusted-advisor-check-refresh-statuses

O código de exemplo a seguir mostra como usar `describe-trusted-advisor-check-refresh-statuses`.

AWS CLI

Para listar os status de atualização das verificações do AWS Trusted Advisor

O `describe-trusted-advisor-check-refresh-statuses` exemplo a seguir lista os status de atualização de duas verificações do Trusted Advisor: Amazon S3 Bucket Permissions and Use. IAM

```
aws support describe-trusted-advisor-check-refresh-statuses \
  --check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

Saída:

```
{
  "statuses": [
    {
      "checkId": "Pfx0RwqBli",
      "status": "none",
      "millisUntilNextRefreshable": 0
    },
    {
      "checkId": "zXCkfM1nI3",
      "status": "none",
      "millisUntilNextRefreshable": 0
    }
  ]
}
```

Para obter mais informações, consulte [AWS Trusted Advisor](#) no AWS Support User Guide.

- Para API obter detalhes, consulte [DescribeTrustedAdvisorCheckRefreshStatuses](#) na Referência de AWS CLI Comandos.

describe-trusted-advisor-check-result

O código de exemplo a seguir mostra como usar `describe-trusted-advisor-check-result`.

AWS CLI

Para listar os resultados de uma verificação do AWS Trusted Advisor

O `describe-trusted-advisor-check-result` exemplo a seguir lista os resultados da verificação de IAM uso.

```
aws support describe-trusted-advisor-check-result \
  --check-id "zXCkfM1nI3"
```

Saída:

```
{
  "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
      }
    ]
  }
}
```

Para obter mais informações, consulte [AWS Trusted Advisor](#) no AWS Support User Guide.

- Para API obter detalhes, consulte [DescribeTrustedAdvisorCheckResult](#) na Referência de AWS CLI Comandos.

describe-trusted-advisor-check-summaries

O código de exemplo a seguir mostra como usar `describe-trusted-advisor-check-summaries`.

AWS CLI

Para listar os resumos das verificações do AWS Trusted Advisor

O `describe-trusted-advisor-check-summaries` exemplo a seguir lista os resultados de duas verificações do Trusted Advisor: Amazon S3 Bucket Permissions and IAM Use.

```
aws support describe-trusted-advisor-check-summaries \
  --check-ids "Pfx0RwqBli" "zXckfM1nI3"
```

Saída:

```
{
  "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
        }
    }
}
]
}

```

Para obter mais informações, consulte [AWS Trusted Advisor](#) no AWS Support User Guide.

- Para API obter detalhes, consulte [DescribeTrustedAdvisorCheckSummaries](#) na Referência de AWS CLI Comandos.

describe-trusted-advisor-checks

O código de exemplo a seguir mostra como usar `describe-trusted-advisor-checks`.

AWS CLI

Para listar as verificações disponíveis do AWS Trusted Advisor

O `describe-trusted-advisor-checks` exemplo a seguir lista os cheques do Trusted Advisor disponíveis em sua AWS conta. Essas informações incluem o nome, ID, descrição, categoria e metadados do cheque. Observe que a saída é reduzida para facilitar a leitura.

```
aws support describe-trusted-advisor-checks \
  --language "en"
```

Saída:

```
{
  "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": []
    }
]
}

```

Para obter mais informações, consulte [AWS Trusted Advisor](#) no AWS Support User Guide.

- Para API obter detalhes, consulte [DescribeTrustedAdvisorChecks](#) na Referência de AWS CLI Comandos.

refresh-trusted-advisor-check

O código de exemplo a seguir mostra como usar `refresh-trusted-advisor-check`.

AWS CLI

Para atualizar uma verificação do AWS Trusted Advisor

O `refresh-trusted-advisor-check` exemplo a seguir atualiza o cheque do Amazon S3 Bucket Permissions Trusted Advisor em AWS sua conta.

```
aws support refresh-trusted-advisor-check \
  --check-id "Pfx0RwqBli"
```

Saída:

```
{
  "status": {
    "checkId": "Pfx0RwqBli",
    "status": "enqueued",

```

```
    "millisUntilNextRefreshable": 3599992
  }
}
```

Para obter mais informações, consulte [AWS Trusted Advisor](#) no AWS Support User Guide.

- Para API obter detalhes, consulte [RefreshTrustedAdvisorCheck](#) na Referência de AWS CLI Comandos.

resolve-case

O código de exemplo a seguir mostra como usar `resolve-case`.

AWS CLI

Como solucionar um caso de suporte

O `resolve-case` exemplo a seguir resolve um caso de suporte em sua AWS conta.

```
aws support resolve-case \
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

Saída:

```
{
  "finalCaseStatus": "resolved",
  "initialCaseStatus": "work-in-progress"
}
```

Para obter mais informações, consulte [Case management](#) no Guia do usuário do AWS Support.

- Para API obter detalhes, consulte [ResolveCase](#) na Referência de AWS CLI Comandos.

SWF Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a AmazonSWF.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

count-closed-workflow-executions

O código de exemplo a seguir mostra como usar `count-closed-workflow-executions`.

AWS CLI

Contando execuções fechadas de fluxo de trabalho

Você pode usar `swf count-closed-workflow-executions` para recuperar o número de execuções de fluxo de trabalho fechadas para um determinado domínio. Você pode especificar filtros para contar classes específicas de execuções.

Os `--start-time-filter` argumentos “`--domain`” `--close-time-filter` “ou” são obrigatórios. Todos os outros argumentos são opcionais.

```
aws swf count-closed-workflow-executions \  
  --domain DataFrobtzz \  
  --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :  
1370044800 }"
```

Saída:

```
{  
  "count": 2,  
  "truncated": false  
}
```

Se “truncado” for `true`, “contagem” representará o número máximo que pode ser devolvido pela Amazon. SWF Quaisquer resultados adicionais são truncados.

Para reduzir o número de resultados retornados, você pode:

modifique os `--start-time-filter` valores `--close-time-filter` ou para restringir o intervalo de tempo pesquisado. Cada um deles é mutuamente exclusivo: você pode especificar somente um deles em uma solicitação. Use os argumentos `--close-status-filter`, `--execution-filter`, `--tag-filter` ou `--type-filter` para filtrar ainda mais os resultados. No entanto, esses argumentos também são mutuamente exclusivos.

Veja também [CountClosedWorkflowExecutions](#) na API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [CountClosedWorkflowExecutions](#) na Referência de AWS CLI Comandos.

count-open-workflow-executions

O código de exemplo a seguir mostra como usar `count-open-workflow-executions`.

AWS CLI

Contando execuções de fluxo de trabalho aberto

Você pode usar `swf count-open-workflow-executions` para recuperar o número de execuções de fluxo de trabalho abertas para um determinado domínio. Você pode especificar filtros para contar classes específicas de execuções.

Os `--start-time-filter` argumentos `--domain` e são obrigatórios. Todos os outros argumentos são opcionais.

```
aws swf count-open-workflow-executions \  
  --domain DataFrobtzz \  
  --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :  
1370044800 }"
```

Saída:

```
{  
  "count": 4,  
  "truncated": false  
}
```

Se “truncado” for `true`, “contagem” representará o número máximo que pode ser devolvido pela Amazon. SWF Quaisquer resultados adicionais são truncados.

Para reduzir o número de resultados retornados, você pode:

modifique os `--start-time-filter` valores para restringir o intervalo de tempo pesquisado. Use os `--type-filter` argumentos `--close-status-filter`, `--execution-filter`, `--tag-filter` ou para filtrar ainda mais os resultados. Cada um deles é mutuamente exclusivo: você pode especificar somente um deles em uma solicitação.

Para obter mais informações, consulte `CountOpenWorkflowExecutions` a API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [CountOpenWorkflowExecutions](#) na Referência de AWS CLI Comandos.

deprecate-domain

O código de exemplo a seguir mostra como usar `deprecate-domain`.

AWS CLI

Depreciando um domínio

Para defasar um domínio (você ainda pode vê-lo, mas não pode criar execuções de fluxo de trabalho ou registrar tipos nele), use `swf deprecate-domain`. Ele tem um único parâmetro obrigatório, `--name`, que causa a reprovação do nome do domínio.

```
aws swf deprecate-domain \  
  --name MyNeatNewDomain ""
```

Assim como ocorre com `register-domain`, nenhuma saída é retornada. No entanto, se você usar `list-domains` para visualizar os domínios registrados, verá que o domínio foi descontinuado e não aparece mais nos dados retornados.

```
aws swf list-domains \  
  --registration-status REGISTERED  
  {  
    "domainInfos": [  
      {  
        "status": "REGISTERED",  
        "name": "DataFrobotz"  
      },  
      {
```

```

    "status": "REGISTERED",
    "name": "erontest"
  }
]
}

```

Se você usar `--registration-status DEPRECATED` com `list-domains`, você verá seu domínio obsoleto.

```

aws swf list-domains \
  --registration-status DEPRECATED
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}

```

Você ainda pode usar `describe-domain` para obter informações sobre um domínio obsoleto.

```

aws swf describe-domain \
  --name MyNeatNewDomain
{
  "domainInfo": {
    "status": "DEPRECATED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}

```

Veja também [DeprecateDomain](#) na API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [DeprecateDomain](#) na Referência de AWS CLI Comandos.

describe-domain

O código de exemplo a seguir mostra como usar `describe-domain`.

AWS CLI

Obtendo informações sobre um domínio

Para obter informações detalhadas sobre um domínio específico, use o `swf describe-domain` comando. Há um parâmetro obrigatório: `--name`, que leva o nome do domínio sobre o qual você deseja informações.

```
aws swf describe-domain \  
  --name DataFrobotz \  
  {  
    "domainInfo": {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "1"  
    }  
  }
```

Você também pode usar `describe-domain` para obter informações sobre domínios obsoletos.

```
aws swf describe-domain \  
  --name MyNeatNewDomain \  
  {  
    "domainInfo": {  
      "status": "DEPRECATED",  
      "name": "MyNeatNewDomain"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "0"  
    }  
  }
```

Veja também [DescribeDomain](#) na API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [DescribeDomain](#) na Referência de AWS CLI Comandos.

list-activity-types

O código de exemplo a seguir mostra como usar `list-activity-types`.

AWS CLI

Listando tipos de atividades

Para obter uma lista dos tipos de atividades de um domínio, use `swf list-activity-types`. Os `--registration-status` argumentos `--domain` e `--registration-status` são obrigatórios.

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED
```

Saída:

```
{
  "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"
  }
]
}

```

Você pode usar o `--name` argumento para selecionar somente tipos de atividade com um nome específico:

```

aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --name "send-subscription-success"

```

Saída:

```

{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.909,
      "activityType": {
        "version": "1",
        "name": "send-subscription-success"
      },
      "description": "subscribe send-subscription-success activity"
    }
  ]
}

```

```
}

```

Para recuperar resultados em páginas, você pode definir o `--maximum-page-size` argumento. Se forem retornados mais resultados do que cabem em uma página de resultados, um `nextPageToken` "" será retornado no conjunto de resultados:

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --maximum-page-size 2

```

Saída:

```
{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLD17QNY7C30pHo9Ssz06D/GuFz10yC73umBQ1t0PJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarki qpSY1ZVveBasBvlvyU
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"
    }
  ]
}
```

Você pode passar o nextPageToken valor para a próxima chamada list-activity-types no --next-page-token argumento, recuperando a próxima página de resultados:

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --maximum-page-size 2 \
  --next-page-token "AAAAKgAAAAEAAAAAAAAAA1Gp1BeLJq
+PmHvAnDxJYbup8+0R4LVtbXLDL7QNY7C30pHo9Sz06D/GuFz10yC73umBQ1t0PJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMLtarki qpSY1ZVveBasBvLvyU
WGAaqehiDz7/JzLT/wWNUM0d+Nhe"
```

Saída:

```
{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCc1gCuq3J+h/
m3+vOfFqtckjLwV5cc40jNAzTCuq/
XcylPumGwkjbajtqPzpbq0cVnFjFxFgoi0LB20lbvv0krbUISBvlpFPmSWpDSZJsxg5UxCcweteS1Fn1PNSZ/
MoinBZo80TkjMuzcsTuK0zH9wCaR8ITcALJ3SaQHU3pyIRS5hPmFA30LIc8zaAepj1aujo6hntNSCruB4"
  "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 ainda houver mais resultados a serem retornados, `nextPageToken` "" será retornado com os resultados. Quando não houver mais páginas de resultados para retornar, "nextPageToken" não será retornado no conjunto de resultados.

Você pode usar o `--reverse-order` argumento para reverter a ordem dos resultados retornados. Isso também afeta os resultados paginados.

```
aws swf list-activity-types \  
  --domain DataFrobtzz \  
  --registration-status REGISTERED \  
  --maximum-page-size 2 \  
  --reverse-order
```

Saída:

```
{  
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAwXcpu5ePSyQkrC  
+8WMBmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCsfGkEn7mk  
+yMCgVc1JxDWmB0EH46bhcmclmYNQihMDmUwopr7To6/R7CLu0St1gkFayx0idJXErQW0zdNfQaIWAnF/  
cwioBbXlkz1fQzmDeU3M5oYGMPQIrUqkPq7pMEW0q0lK5eDN97NzFYdZZ/rlcLDWPZhUjY",  
  "typeInfos": [  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.085,  
      "activityType": {  
        "version": "1",  
        "name": "subscribe-user-sns"  
      },  
      "description": "subscribe subscribe-user-sns activity"  
    },  
    {  
      "status": "REGISTERED",  
      "creationDate": 1371454150.909,  
      "activityType": {  
        "version": "1",  
        "name": "send-subscription-success"  
      },  
      "description": "subscribe send-subscription-success activity"  
    }  
  ]  
}
```

Veja também [ListActivityTypes](#) na APIReferência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [ListActivityTypes](#) na Referência de AWS CLI Comandos.

list-domains

O código de exemplo a seguir mostra como usar `list-domains`.

AWS CLI

Exemplo 1: Para listar seus domínios registrados

O exemplo de `list-domains` comando a seguir lista os REGISTERED SWF domínios que você registrou para sua conta.

```
aws swf list-domains \  
  --registration-status REGISTERED
```

Saída:

```
{  
  "domainInfos": [  
    {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    },  
    {  
      "status": "REGISTERED",  
      "name": "erontest"  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListDomains](#) na APIReferência do Amazon Simple Workflow Service

Exemplo 2: Para listar seus domínios obsoletos

O exemplo de `list-domains` comando a seguir lista os DEPRECATED SWF domínios que você registrou para sua conta. Domínios obsoletos são domínios que não podem registrar novos fluxos de trabalho ou atividades, mas que ainda podem ser consultados.

```
aws swf list-domains \  
  --registration-status DEPRECATED
```

Saída:

```
{  
  "domainInfos": [  
    {  
      "status": "DEPRECATED",  
      "name": "MyNeatNewDomain"  
    }  
  ]  
}
```

Para obter mais informações, consulte [ListDomains](#) a API Referência do Amazon Simple Workflow Service

Exemplo 3: Para listar a primeira página de domínios registrados

O exemplo de `list-domains` comando a seguir lista os REGISTERED SWF domínios da primeira página que você registrou em sua conta usando a `--maximum-page-size` opção.

```
aws swf list-domains \  
  --registration-status REGISTERED \  
  --maximum-page-size 1
```

Saída:

```
{  
  "domainInfos": [  
    {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    }  
  ],  
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAA2QJKNtidVgd49TTeNwYcpD  
+QKT2ynuEbibcQWe2QKrs1MGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it  
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKE0Pm3k2r40jAGaFsG0uVbrK1jv1a7wdU7FYH301kNCP8b7PBj9SBkUyGoiAg  
}
```

Para obter mais informações, consulte [ListDomains](#) na APIReferência do Amazon Simple Workflow Service

Exemplo 4: Para listar a página única especificada de domínios registrados

O exemplo de `list-domains` comando a seguir lista os REGISTERED SWF domínios da primeira página que você registrou em sua conta usando a `--maximum-page-size` opção.

Quando você fizer a chamada novamente, desta vez fornecendo o valor de `nextPageToken` no `--next-page-token` argumento, você obterá outra página de resultados.

```
aws swf list-domains \
  --registration-status REGISTERED \
  --maximum-page-size 1 \
  --next-page-token "AAAAKgAAAAEAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbIBCQWe2QKrsLMGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdzIztvIxIKEOPm3k2r40jAGaFsG0uVbrKljvLa7wdU7FYH30lkNCP8b7PBj9SBkUyGoiAg
```

Saída:

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

Quando não há mais páginas de resultados para recuperar, `nextPageToken` não será retornado nos resultados.

Para obter mais informações, consulte [ListDomains](#) na APIReferência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [ListDomains](#) na Referência de AWS CLI Comandos.

list-workflow-types

O código de exemplo a seguir mostra como usar `list-workflow-types`.

AWS CLI

Listando tipos de fluxo de trabalho

Para obter uma lista dos tipos de fluxo de trabalho de um domínio, use `swf list-workflow-types`. Os `--registration-status` argumentos `--domain` e são obrigatórios. Aqui está um exemplo simples.

```
aws swf list-workflow-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED
```

Saída:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.598,
      "description": "DataFrobtzz subscribe workflow",
      "workflowType": {
        "version": "v3",
        "name": "subscribe"
      }
    }
  ]
}
```

Da mesma forma `list-activity-types`, você pode usar o `--name` argumento para selecionar somente tipos de fluxo de trabalho com um nome específico e usar o `--maximum-page-size` argumento em coordenação com `--next-page-token` os resultados da página. Para reverter a ordem em que os resultados são retornados, use `--reverse-order`.

Veja também [ListWorkflowTypes](#) na API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [ListWorkflowTypes](#) na Referência de AWS CLI Comandos.

register-domain

O código de exemplo a seguir mostra como usar `register-domain`.

AWS CLI

Registrando um domínio

Você pode usar o AWS CLI para registrar novos domínios. Use o comando `swf register-domain`. Há dois parâmetros obrigatórios, `--name`, que usam o nome do domínio e `--workflow-execution-retention-period-in-days`, que usam um número inteiro para especificar o número de dias para reter os dados de execução do fluxo de trabalho nesse domínio, até um período máximo de 90 dias (para obter mais informações, consulte SWF FAQ < https://aws.amazon.com/swf/faqs/#retain_limit >). Os dados de execução do fluxo de trabalho não serão retidos após o período especificado de dias.

```
aws swf register-domain \  
  --name MyNeatNewDomain \  
  --workflow-execution-retention-period-in-days 0  
  ""
```

Ao registrar um domínio, nada será retornado (""), mas você pode usar `swf list-domains` ou `swf describe-domain` para ver o novo domínio.

```
aws swf list-domains \  
  --registration-status REGISTERED  
  {  
    "domainInfos": [  
      {  
        "status": "REGISTERED",  
        "name": "DataFrobotz"  
      },  
      {  
        "status": "REGISTERED",  
        "name": "MyNeatNewDomain"  
      },  
      {  
        "status": "REGISTERED",  
        "name": "erontest"  
      }  
    ]  
  }
```

Usar `swf describe-domain`:

```
aws swf describe-domain --
name MyNeatNewDomain
{
  "domainInfo": {
    "status": "REGISTERED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}
```

Veja também [RegisterDomain](#) na API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [RegisterDomain](#) na Referência de AWS CLI Comandos.

register-workflow-type

O código de exemplo a seguir mostra como usar `register-workflow-type`.

AWS CLI

Registrando um tipo de fluxo de trabalho

Para registrar um tipo de fluxo de trabalho com o AWS CLI, use o `swf register-workflow-type` comando.

```
aws swf register-workflow-type \
  --domain DataFrobtzz \
  --name "MySimpleWorkflow" \
  --workflow-version "v1"
```

Se for bem-sucedido, o comando não produzirá nenhuma saída.

Em caso de erro (por exemplo, se você tentar registrar o mesmo fluxo de trabalho duas vezes ou especificar um domínio que não existe), você receberá uma resposta em JSON

```
{
  "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
  "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
}
```

Os `--domain`, `--name` e `--workflow-version` são obrigatórios. Você também pode definir a descrição do fluxo de trabalho, os tempos limite e a política de fluxo de trabalho secundário.

Para obter mais informações, consulte [RegisterWorkflowType](#) a API Referência do Amazon Simple Workflow Service

- Para API obter detalhes, consulte [RegisterWorkflowType](#) na Referência de AWS CLI Comandos.

Exemplos do Systems Manager usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Systems Manager.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

add-tags-to-resource

O código de exemplo a seguir mostra como usar `add-tags-to-resource`.

AWS CLI

Exemplo 1: adicionar tags a uma janela de manutenção

O exemplo de `add-tags-to-resource` a seguir adiciona uma tag à janela de manutenção especificada.

```
aws ssm add-tags-to-resource \  
  --resource-type "MaintenanceWindow" \  
  --resource-id "mw-03eb9db428EXAMPLE" \  
  --tags "Key=Stack,Value=Production"
```



```
--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"
```

Saída:

```
{  
  "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"  
}
```

Para obter mais informações, consulte [Trabalhando com incidentes do Incident Manager OpsCenter no Guia do Usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [AssociateOpsItemRelatedItem](#) na Referência de AWS CLI Comandos.

cancel-command

O código de exemplo a seguir mostra como usar `cancel-command`.

AWS CLI

Exemplo 1: cancelar um comando para todas as instâncias

O exemplo de `cancel-command` a seguir tenta cancelar o comando especificado que já está em execução para todas as instâncias.

```
aws ssm cancel-command \  
--command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

Este comando não produz saída.

Exemplo 2: cancelar um comando para instâncias específicas

O exemplo de `cancel-command` a seguir tenta cancelar um comando somente para a instância especificada.

```
aws ssm cancel-command \  
--command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"  
--instance-ids "i-02573cafcfEXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Marcar parâmetros do Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [CancelCommand](#) na Referência de AWS CLI Comandos.

cancel-maintenance-window-execution

O código de exemplo a seguir mostra como usar `cancel-maintenance-window-execution`.

AWS CLI

Para cancelar a execução de uma janela de manutenção

Este `cancel-maintenance-window-execution` exemplo interrompe a execução da janela de manutenção especificada que já está em andamento.

```
aws ssm cancel-maintenance-window-execution \  
  --window-execution-id j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

Saída:

```
{  
  "WindowExecutionId": "j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"  
}
```

Para obter mais informações, consulte [Systems Manager Maintenance Windows Tutorials \(AWS CLI\)](#) no AWS Systems Manager User Guide.

- Para API obter detalhes, consulte [CancelMaintenanceWindowExecution](#) na Referência de AWS CLI Comandos.

create-activation

O código de exemplo a seguir mostra como usar `create-activation`.

AWS CLI

Para criar uma ativação de instância gerenciada

O exemplo de `create-activation` a seguir cria uma ativação de instância gerenciada.

```
aws ssm create-activation \  
  --default-instance-name "HybridWebServers" \  
  --iam-role "HybridWebServersRole" \  
  --registration-limit 5
```

Saída:

```
{  
  "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",  
  "ActivationCode": "dRmgnYaFv567vEXAMPLE"  
}
```

Para obter mais informações, consulte [Etapa 4: criar uma ativação de instância gerenciada para um ambiente híbrido](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [CreateActivation](#) na Referência de AWS CLI Comandos.

create-association-batch

O código de exemplo a seguir mostra como usar `create-association-batch`.

AWS CLI

Para criar várias associações

Este exemplo associa um documento de configuração a várias instâncias. A saída retorna uma lista de operações bem e malsucedidas, se aplicável.

Comando:

```
aws ssm create-association-batch --entries "Name=AWS-  
UpdateSSMAgent, InstanceId=i-1234567890abcdef0" "Name=AWS-  
UpdateSSMAgent, InstanceId=i-9876543210abcdef0"
```

Saída:

```
{  
  "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": []
}

```

- Para API obter detalhes, consulte [CreateAssociationBatch](#) na Referência de AWS CLI Comandos.

create-association

O código de exemplo a seguir mostra como usar create-association.

AWS CLI

Exemplo 1: Para associar um documento usando a instância IDs

Este exemplo associa um documento de configuração a uma instância, usando instanceIDs.

```

aws ssm create-association \
  --instance-id "i-0cb2b964d3e14fd9f" \
  --name "AWS-UpdateSSMAgent"

```

Saída:

```

{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [CreateAssociation](#) na API Referência do AWS Systems Manager.

Exemplo 2: associar um documento usando destinos

Esse exemplo associa um documento de configuração a uma instância usando destinos.

```

aws ssm create-association \
  --name "AWS-UpdateSSMAgent" \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"

```

Saída:

```

{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [CreateAssociation](#) na API Referência do AWS Systems Manager.

Exemplo 3: criar uma associação para ser executada somente uma vez

Esse exemplo cria uma nova associação que só é executada uma vez na data e na hora especificadas. As associações criadas com uma data no passado ou no presente (no momento em que são processadas, a data está no passado) são executadas imediatamente.

```

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

```

Saída:

```

{
  "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"
      }
    ]
  }
}

```

Para obter mais informações, consulte [CreateAssociation](#) na API Referência ou Referência do AWS Systems Manager: [expressões Cron e de taxa para Systems Manager](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [CreateAssociation](#) na Referência de AWS CLI Comandos.

create-document

O código de exemplo a seguir mostra como usar create-document.

AWS CLI

Para criar um documento

O exemplo de create-document a seguir cria um documento do Systems Manager.

```

aws ssm create-document \
  --content file://exampleDocument.yml \
  --name "Example" \
  --document-type "Automation" \
  --document-format YAML

```

Saída:

```

{
  "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": []
  }
}

```

Para obter mais informações, consulte [Criar documentos do Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [CreateDocument](#) na Referência de AWS CLI Comandos.

create-maintenance-window

O código de exemplo a seguir mostra como usar `create-maintenance-window`.

AWS CLI

Exemplo 1: criar uma janela de manutenção

O exemplo de `create-maintenance-window` a seguir cria uma nova janela de manutenção que, a cada cinco minutos, e por até duas horas (conforme necessário), impede que novas tarefas sejam iniciadas dentro de uma hora do final da execução da janela de manutenção, permite destinos não associados (instâncias que você não registrou na janela de manutenção) e indica, por meio do uso de tags personalizadas, que seu criador pretende usá-la em um 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"
```

Saída:

```
{  
  "WindowId": "mw-0c50858d01EXAMPLE"  
}
```

Exemplo 2: criar uma janela de manutenção que é executada somente uma vez

O exemplo de `create-maintenance-window` a seguir cria uma nova janela de manutenção que só é executada uma vez na data e na hora especificadas.

```
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"
```

Saída:

```
{  
  "WindowId": "mw-01234567890abcdef"  
}
```

Para obter mais informações, consulte [Janelas de manutenção](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [CreateMaintenanceWindow](#) na Referência de AWS CLI Comandos.

create-ops-item

O código de exemplo a seguir mostra como usar `create-ops-item`.

AWS CLI

Para criar um OpsItems

O `create-ops-item` exemplo a seguir usa a chave `/aws/resources OperationalData` para criar um OpsItem com um recurso relacionado ao 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"
```

Saída:

```
{  
  "OpsItemId": "oi-1a2b3c4d5e6f"  
}
```

Para obter mais informações, consulte [Creating OpsItems](#) in the AWS Systems Manager User Guide.

- Para API obter detalhes, consulte [CreateOpsItem](#) na Referência de AWS CLI Comandos.

create-patch-baseline

O código de exemplo a seguir mostra como usar `create-patch-baseline`.

AWS CLI

Exemplo 1: criar uma lista de referência de patches com aprovação automática

O exemplo de `create-patch-baseline` a seguir cria uma lista de referência de patches para o Windows Server que aprova patches para instâncias de produção sete dias após serem lançados pela 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"
```

Saída:

```
{  
  "BaselineId": "pb-045f10b4f3EXAMPLE"  
}
```

Exemplo 2: criar uma lista de referência de patches com uma data limite para aprovação

O exemplo de `create-patch-baseline` a seguir cria uma lista de referência de patches para o Windows Server que aprova todos os patches para um ambiente de produção lançados até 7 de julho de 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"
```

Saída:

```
{
  "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Exemplo 3: Para criar uma linha de base de patch com regras de aprovação armazenadas em um arquivo JSON

O exemplo de `create-patch-baseline` a seguir cria uma lista de referência de patches para o Amazon Linux 2017.09 que aprova patches para um ambiente de produção sete dias após seu lançamento, especifica regras de aprovação para a lista de referência de patches e especifica um repositório personalizado para patches.

```
aws ssm create-patch-baseline \
  --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Conteúdo de `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"
    }
]
}

```

Exemplo 4: criar uma lista de referência de patches que especifica patches aprovados e rejeitados

O exemplo de `create-patch-baseline` a seguir especifica explicitamente os patches a serem aprovados e rejeitados como exceção às regras de aprovação padrão.

```

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"
```

Para obter mais informações, consulte [Criar uma lista de referência de patches personalizada](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [CreatePatchBaseline](#) na Referência de AWS CLI Comandos.

create-resource-data-sync

O código de exemplo a seguir mostra como usar `create-resource-data-sync`.

AWS CLI

Para criar uma sincronização de dados de recursos

Este exemplo cria uma sincronização de dados de recursos. Não haverá saída se o comando for bem-sucedido.

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"
```

- Para API obter detalhes, consulte [CreateResourceDataSync](#) na Referência de AWS CLI Comandos.

delete-activation

O código de exemplo a seguir mostra como usar `delete-activation`.

AWS CLI

Para excluir uma ativação de instância gerenciada

O exemplo de `delete-activation` a seguir exclui uma ativação de instância gerenciada.

```
aws ssm delete-activation \  

```

```
--activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando o AWS Systems Manager para ambientes híbridos](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeleteActivation](#) na Referência de AWS CLI Comandos.

delete-association

O código de exemplo a seguir mostra como usar delete-association.

AWS CLI

Exemplo 1: excluir uma associação usando o ID da associação

O exemplo de delete-association a seguir exclui a associação para o ID de associação especificado. Não haverá saída se o comando for bem-sucedido.

```
aws ssm delete-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Este comando não produz saída.

Para obter mais informações, consulte [Editar e criar uma nova versão de uma associação](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: excluir uma associação

O exemplo de delete-association a seguir exclui a associação entre uma instância e um documento. Não haverá saída se o comando for bem-sucedido.

```
aws ssm delete-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhar com associações no Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeleteAssociation](#) na Referência de AWS CLI Comandos.

delete-document

O código de exemplo a seguir mostra como usar delete-document.

AWS CLI

Para excluir um documento

O exemplo de delete-document a seguir exclui um documento do Systems Manager.

```
aws ssm delete-document \  
  --name "Example"
```

Este comando não produz saída.

Para obter mais informações, consulte [Criar documentos do Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeleteDocument](#) na Referência de AWS CLI Comandos.

delete-inventory

O código de exemplo a seguir mostra como usar delete-inventory.

AWS CLI

Para excluir um tipo de inventário personalizado

Este exemplo exclui um esquema de inventário personalizado.

Comando:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DeleteSchema"
```

Saída:

```
{  
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",  
  "TypeName": "Custom:RackInfo",  
  "DeletionSummary": {  
    "TotalCount": 1,  
    "RemainingCount": 1,  
  }  
}
```

```

    "SummaryItems": [
      {
        "Version": "1.0",
        "Count": 1,
        "RemainingCount": 1
      }
    ]
  }
}

```

Para desativar um tipo de inventário personalizado

Este exemplo desativa um esquema de inventário personalizado.

Comando:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-option "DisableSchema"
```

Saída:

```

{
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",
  "TypeName": "Custom:RackInformation",
  "DeletionSummary": {
    "TotalCount": 0,
    "RemainingCount": 0,
    "SummaryItems": []
  }
}

```

- Para API obter detalhes, consulte [DeleteInventory](#) na Referência de AWS CLI Comandos.

delete-maintenance-window

O código de exemplo a seguir mostra como usar delete-maintenance-window.

AWS CLI

Para excluir uma janela de manutenção

Este exemplo de delete-maintenance-window remove a janela de manutenção especificada.

```
aws ssm delete-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9"
```

Saída:

```
{  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9"  
}
```

Para obter mais informações, consulte [Excluir uma janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeleteMaintenanceWindow](#) na Referência de AWS CLI Comandos.

delete-parameter

O código de exemplo a seguir mostra como usar `delete-parameter`.

AWS CLI

Para excluir um parâmetro

O exemplo de `delete-parameter` a seguir exclui o parâmetro único especificado.

```
aws ssm delete-parameter \  
  --name "MyParameter"
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhar com o Parameter Store](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeleteParameter](#) na Referência de AWS CLI Comandos.

delete-parameters

O código de exemplo a seguir mostra como usar `delete-parameters`.

AWS CLI

Para excluir uma lista de parâmetros

O `delete-parameters` exemplo a seguir exclui os parâmetros especificados.

```
aws ssm delete-parameters \  
  --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

Saída:

```
{  
  "DeletedParameters": [  
    "MyFirstParameter",  
    "MySecondParameter"  
  ],  
  "InvalidParameters": [  
    "MyInvalidParameterName"  
  ]  
}
```

Para obter mais informações, consulte [Trabalhar com o Parameter Store](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeleteParameters](#) na Referência de AWS CLI Comandos.

`delete-patch-baseline`

O código de exemplo a seguir mostra como usar `delete-patch-baseline`.

AWS CLI

Para excluir uma lista de referência de patches

O exemplo de `delete-patch-baseline` a seguir exclui a lista de referência de patches especificada.

```
aws ssm delete-patch-baseline \  
  --baseline-id "pb-045f10b4f382baeda"
```

Saída:

```
{  
  "BaselineId": "pb-045f10b4f382baeda"  
}
```

Para obter mais informações, consulte [Atualizar ou excluir uma lista de referência de patches \(Console\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeletePatchBaseline](#) na Referência de AWS CLI Comandos.

delete-resource-data-sync

O código de exemplo a seguir mostra como usar `delete-resource-data-sync`.

AWS CLI

Para excluir uma sincronização de dados de recursos

Este exemplo exclui uma sincronização de dados de recursos. Não haverá saída se o comando for bem-sucedido.

Comando:

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- Para API obter detalhes, consulte [DeleteResourceDataSync](#) na Referência de AWS CLI Comandos.

deregister-managed-instance

O código de exemplo a seguir mostra como usar `deregister-managed-instance`.

AWS CLI

Para cancelar o registro de uma instância gerenciada

O exemplo de `deregister-managed-instance` a seguir cancela o registro da instância gerenciada especificada.

```
aws ssm deregister-managed-instance  
--instance-id "mi-08ab247cdfEXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Cancelar o registro de instâncias gerenciadas em ambientes híbridos](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeregisterManagedInstance](#) na Referência de AWS CLI Comandos.

deregister-patch-baseline-for-patch-group

O código de exemplo a seguir mostra como usar `deregister-patch-baseline-for-patch-group`.

AWS CLI

Para cancelar o registro de um grupo de patches de uma lista de referência de patches

O exemplo de `deregister-patch-baseline-for-patch-group` a seguir cancela o registro do grupo de patches especificado da lista de referência de patches especificada.

```
aws ssm deregister-patch-baseline-for-patch-group \  
  --patch-group "Production" \  
  --baseline-id "pb-0ca44a362fEXAMPLE"
```

Saída:

```
{  
  "PatchGroup": "Production",  
  "BaselineId": "pb-0ca44a362fEXAMPLE"  
}
```

Para obter mais informações, consulte [Adicionar um grupo de patches a uma lista de referência de patches](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeregisterPatchBaselineForPatchGroup](#) na Referência de AWS CLI Comandos.

deregister-target-from-maintenance-window

O código de exemplo a seguir mostra como usar `deregister-target-from-maintenance-window`.

AWS CLI

Para remover um destino de uma janela de manutenção

O exemplo de `deregister-target-from-maintenance-window` a seguir remove o destino especificado da janela de manutenção especificada.

```
aws ssm deregister-target-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Saída:

```
{  
  "WindowId": "mw-ab12cd34ef56gh78",  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Para obter mais informações, consulte [Atualizar uma janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DeregisterTargetFromMaintenanceWindow](#) na Referência de AWS CLI Comandos.

deregister-task-from-maintenance-window

O código de exemplo a seguir mostra como usar `deregister-task-from-maintenance-window`.

AWS CLI

Para remover uma tarefa de uma janela de manutenção

O exemplo de `deregister-task-from-maintenance-window` a seguir remove a tarefa especificada da janela de manutenção especificada.

```
aws ssm deregister-task-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Saída:

```
{  
  "WindowTaskId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",  
  "WindowId": "mw-ab12cd34ef56gh78"  
}
```

```
}
```

Para obter mais informações, consulte [Systems Manager Maintenance Windows Tutorials \(AWS CLI\)](#) no AWS Systems Manager User Guide.

- Para API obter detalhes, consulte [DeregisterTaskFromMaintenanceWindow](#) na Referência de AWS CLI Comandos.

describe-activations

O código de exemplo a seguir mostra como usar `describe-activations`.

AWS CLI

Para descrever as ativações

O `describe-activations` exemplo a seguir lista detalhes sobre as ativações em sua AWS conta.

```
aws ssm describe-activations
```

Saída:

```
{
  "ActivationList": [
    {
      "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
      "Description": "Example1",
      "IamRole": "HybridWebServersRole",
      "RegistrationLimit": 5,
      "RegistrationsCount": 5,
      "ExpirationDate": 1584316800.0,
      "Expired": false,
      "CreatedDate": 1581954699.792
    },
    {
      "ActivationId": "3ee0322b-f62d-40eb-b672-13ebfEXAMPLE",
      "Description": "Example2",
      "IamRole": "HybridDatabaseServersRole",
      "RegistrationLimit": 5,
      "RegistrationsCount": 5,
      "ExpirationDate": 1580515200.0,
      "Expired": true,

```

```

        "CreateDate": 1578064132.002
      },
    ]
  }

```

Para obter mais informações, consulte [Etapa 4: criar uma ativação de instância gerenciada para um ambiente híbrido](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeActivations](#) na Referência de AWS CLI Comandos.

describe-association-execution-targets

O código de exemplo a seguir mostra como usar `describe-association-execution-targets`.

AWS CLI

Para obter detalhes da execução de uma associação

O exemplo de `describe-association-execution-targets` a seguir descreve a execução da associação especificada.

```

aws ssm describe-association-execution-targets \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"

```

Saída:

```

{
  "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"
      }
    }
  ]
}

```

```
]
}
```

Para obter mais informações, consulte [Visualizar históricos de associações](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeAssociationExecutionTargets](#) na Referência de AWS CLI Comandos.

describe-association-executions

O código de exemplo a seguir mostra como usar `describe-association-executions`.

AWS CLI

Exemplo 1: obter detalhes de todas as execuções de uma associação

O exemplo de `describe-association-executions` a seguir descreve todas as execuções da associação especificada.

```
aws ssm describe-association-executions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Saída:

```
{
  "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}"
    },
    ...
]
}

```

Para obter mais informações, consulte [Visualizar históricos de associações](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: obter detalhes de todas as execuções de uma associação após uma data e uma hora específicas

O exemplo de `describe-association-executions` a seguir descreve todas as execuções de uma associação após a data e a hora especificadas.

```

aws ssm describe-association-executions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"

```

Saída:

```

{
  "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}"
    },
    ...
  ]
}

```



```
}
```

Para obter mais informações, consulte [Visualizar históricos de associações](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeAssociationExecutions](#) na Referência de AWS CLI Comandos.

describe-association

O código de exemplo a seguir mostra como usar describe-association.

AWS CLI

Exemplo 1: obter detalhes de uma associação

O exemplo de describe-association a seguir descreve a associação para o ID de associação especificado.

```
aws ssm describe-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Saída:

```
{  
  "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"
}
}
```

Para obter mais informações, consulte [Editar e criar uma nova versão de uma associação](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: obter detalhes de uma associação para uma instância e um documento específicos

O exemplo de `describe-association` a seguir descreve a associação entre uma instância e um documento.

```
aws ssm describe-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

Saída:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```
}
```

Para obter mais informações, consulte [Editar e criar uma nova versão de uma associação](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeAssociation](#) na Referência de AWS CLI Comandos.

describe-automation-executions

O código de exemplo a seguir mostra como usar `describe-automation-executions`.

AWS CLI

Para descrever uma execução do Automation

O exemplo de `describe-automation-executions` a seguir exibe detalhes sobre uma execução do Automation.

```
aws ssm describe-automation-executions \
  --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Saída:

```
{
  "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"
    }
  ]
}
```

```

    }
  ]
}

```

Para obter mais informações, consulte [Executar um fluxo de trabalho simples do Automation](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeAutomationExecutions](#) na Referência de AWS CLI Comandos.

describe-automation-step-executions

O código de exemplo a seguir mostra como usar `describe-automation-step-executions`.

AWS CLI

Exemplo 1: descrever todas as etapas de uma execução de automação

O exemplo de `describe-automation-step-executions` a seguir exibe detalhes sobre as etapas de uma execução do Automation.

```

aws ssm describe-automation-step-executions \
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE

```

Saída:

```

{
  "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": {}
  }
]
}

```

Exemplo 2: descrever uma etapa específica de uma execução do Automation

O exemplo de `describe-automation-step-executions` a seguir exibe detalhes sobre uma etapa específica de uma execução do Automation.

```

aws ssm describe-automation-step-executions \
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
  --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE

```

Para obter mais informações, consulte [Executar um fluxo de trabalho do Automation passo a passo \(Linha de comando\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeAutomationStepExecutions](#) na Referência de AWS CLI Comandos.

describe-available-patches

O código de exemplo a seguir mostra como usar `describe-available-patches`.

AWS CLI

Para obter os patches disponíveis

O `describe-available-patches` exemplo a seguir recupera detalhes sobre todos os patches disponíveis para o Windows Server 2019 que têm uma MSRC severidade Crítica.

```

aws ssm describe-available-patches \
  --
  filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"

```

Saída:

```

{
  "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"
    },
    ...
]
}

```

Para obter detalhes de um patch específico

O exemplo de `describe-available-patches` a seguir recupera detalhes do patch especificado.

```
aws ssm describe-available-patches \  
--filters "Key=PATCH_ID,Values=KB4480979"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Como as operações do Patch Manager funcionam](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeAvailablePatches](#) na Referência de AWS CLI Comandos.

describe-document-permission

O código de exemplo a seguir mostra como usar `describe-document-permission`.

AWS CLI

Para descrever permissões do documento

O exemplo de `describe-document-permission` a seguir exibe detalhes de permissão sobre um documento do Systems Manager que é compartilhado publicamente.

```
aws ssm describe-document-permission \  
  --name "Example" \  
  --permission-type "Share"
```

Saída:

```
{  
  "AccountIds": [  
    "all"  
  ],  
  "AccountSharingInfoList": [  
    {  
      "AccountId": "all",  
      "SharedDocumentVersion": "$DEFAULT"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Compartilhar um documento do Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeDocumentPermission](#) na Referência de AWS CLI Comandos.

describe-document

O código de exemplo a seguir mostra como usar `describe-document`.

AWS CLI

Para exibir detalhes de um documento

O `describe-document` exemplo a seguir exibe detalhes sobre um documento do Systems Manager em sua AWS conta.

```
aws ssm describe-document \  
  --name "Example"
```

Saída:

```
{  
  "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": []  
  }  
}
```

```
}

```

Para obter mais informações, consulte [Criar documentos do Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeDocument](#) na Referência de AWS CLI Comandos.

describe-effective-instance-associations

O código de exemplo a seguir mostra como usar `describe-effective-instance-associations`.

AWS CLI

Para obter detalhes das associações efetivas de uma instância

O exemplo de `describe-effective-instance-associations` a seguir recupera detalhes sobre as associações efetivas de uma instância.

Comando:

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

Saída:

```
{
  "Associations": [
    {
      "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
      "InstanceId": "i-1234567890abcdef0",
      "Content": "{\n  \"schemaVersion\": \"1.2\",\n  \"description\":\n  \"Update the Amazon SSM Agent to the latest version or specified version.\",\n  \"parameters\": {\n    \"version\": {\n      \"default\": \"\",\n      \"description\": \"(Optional) A specific version of the Amazon SSM Agent\n  to install. If not specified, the agent will be updated to the latest version.\",\n      \"type\": \"String\",\n      \"allowDowngrade\": {\n        \"default\": \"false\",\n        \"description\": \"(Optional)\n  Allow the Amazon SSM Agent service to be downgraded to an earlier version. If\n  set to false, the service can be upgraded to newer versions only (default). If\n  set to true, specify the earlier version.\",\n        \"type\": \"String\",\n        \"allowedValues\": [\n          \"true\",\n          \"false\"\n        ]\n      },\n      \"runtimeConfig\": {\n        \"aws:updateSsmAgent\": {\n          \"properties\": [\n
```

```

        \"agentName\": \"amazon-ssm-agent\",
        \"source\":
        \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-manifest.json\",
        \"allowDowngrade\": \"{{ allowDowngrade }}\",
        \"targetVersion\": \"{{ version }}\"
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeEffectiveInstanceAssociations](#) na Referência de AWS CLI Comandos.

describe-effective-patches-for-patch-baseline

O código de exemplo a seguir mostra como usar `describe-effective-patches-for-patch-baseline`.

AWS CLI

Exemplo 1: obter todos os patches definidos por uma lista de referência de patches

O `describe-effective-patches-for-patch-baseline` exemplo a seguir retorna os patches definidos por uma linha de base de patch personalizada na AWS conta atual. Observe que, para uma lista de referência personalizada, somente a ID é necessária para `--baseline-id`.

```
aws ssm describe-effective-patches-for-patch-baseline \
  --baseline-id "pb-08b654cf9b9681f04"
```

Saída:

```

{
  "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--"
}

```

Exemplo 2: Para obter todos os patches definidos por uma linha de base de patches AWS gerenciados

O `describe-effective-patches-for-patch-baseline` exemplo a seguir retorna os patches definidos por uma linha de base de patches AWS gerenciados. Observe que, para uma linha de base AWS gerenciada, a linha de base completa ARN é necessária para `--baseline-id`

```

aws ssm describe-effective-patches-for-patch-baseline \
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"

```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Como os patches de segurança são selecionados](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeEffectivePatchesForPatchBaseline](#) na Referência de AWS CLI Comandos.

describe-instance-associations-status

O código de exemplo a seguir mostra como usar `describe-instance-associations-status`.

AWS CLI

Para descrever o status das associações de uma instância

Este exemplo mostra detalhes das associações de uma instância.

Comando:

```

aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"

```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeInstanceAssociationsStatus](#) na Referência de AWS CLI Comandos.

describe-instance-information

O código de exemplo a seguir mostra como usar `describe-instance-information`.

AWS CLI

Exemplo 1: descrever as informações da instância gerenciada

O exemplo de `describe-instance-information` a seguir recupera detalhes de cada uma das suas instâncias gerenciadas.

aws ssm describe-instance-information

Exemplo 2: descrever informações sobre uma instância gerenciada específica

O exemplo de `describe-instance-information` a seguir mostra os detalhes da instância gerenciada `i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \  
  --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Exemplo 3: descrever informações sobre instâncias gerenciadas com uma chave de tag específica

O exemplo de `describe-instance-information` a seguir mostra detalhes de instâncias gerenciadas que têm a chave de tag `DEV`.

```
aws ssm describe-instance-information \  
  --filters "Key=tag-key,Values=DEV"
```

Saída:

```
{  
  "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        }  
      }  
    }  
  ]  
}
```



```
}
  }
}
]
```

Para obter mais informações, consulte [Instâncias gerenciadas](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeInstanceInformation](#) na Referência de AWS CLI Comandos.

describe-instance-patch-states-for-patch-group

O código de exemplo a seguir mostra como usar `describe-instance-patch-states-for-patch-group`.

AWS CLI

Exemplo 1: obter os estados da instância de um grupo de patches

O exemplo de `describe-instance-patch-states-for-patch-group` a seguir recupera detalhes sobre os estados resumidos de patches por instância para o grupo de patches especificado.

```
aws ssm describe-instance-patch-states-for-patch-group \
  --patch-group "Production"
```

Saída:

```
{
  "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
  }
]
}

```

Exemplo 2: obter os estados da instância de um grupo de patches com mais de cinco patches ausentes

O exemplo de `describe-instance-patch-states-for-patch-group` a seguir recupera detalhes sobre os estados resumidos de patches para o grupo de patches especificado por instâncias com mais de cinco patches ausentes.

```
aws ssm describe-instance-patch-states-for-patch-group \  
  --filters Key=MissingCount,Type=GreaterThan,Values=5 \  
  --patch-group "Production"
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Exemplo 3: obter os estados da instância de um grupo de patches com menos de dez instâncias que exigem uma reinicialização

O exemplo de `describe-instance-patch-states-for-patch-group` a seguir recupera detalhes sobre os estados resumidos de patches para o grupo de patches especificado por instâncias com menos de dez instâncias que exigem uma reinicialização.

```
aws ssm describe-instance-patch-states-for-patch-group \  
  --filters Key=MissingCount,Type=GreaterThan,Values=5 \  
  --patch-group "Production"
```

```
--filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \  
--patch-group "Production"
```

Saída:

```
{  
  "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  
    }  
  ]  
}
```

Para obter mais informações, consulte [Noções básicas sobre valores de estado de conformidade de patches](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeInstancePatchStatesForPatchGroup](#) na Referência de AWS CLI Comandos.

describe-instance-patch-states

O código de exemplo a seguir mostra como usar `describe-instance-patch-states`.

AWS CLI

Para obter os estados resumidos de patches para instâncias

Este exemplo de `describe-instance-patch-states` obtém os estados resumidos de patches para uma instância.

```
aws ssm describe-instance-patch-states \  
  --instance-ids "i-1234567890abcdef0"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Sobre a conformidade de patches](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeInstancePatchStates](#) na Referência de AWS CLI Comandos.

describe-instance-patches

O código de exemplo a seguir mostra como usar `describe-instance-patches`.

AWS CLI

Exemplo 1: obter os detalhes do estado do patch para uma instância

O exemplo de `describe-instance-patches` a seguir recupera detalhes sobre os patches para a instância especificada.

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0"
```

Saída:

```
{  
  "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--"  
}
```

Exemplo 2: obter uma lista de patches no estado Ausente para uma instância

O exemplo de `describe-instance-patches` a seguir recupera informações sobre patches que estão no estado Ausente para a instância especificada.

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0" \  
  --filters Key=State,Values=Missing
```

Saída:

```
{  
  "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--"  
}
```

Para obter mais informações, consulte [Sobre estados de conformidade de patches](#) no Guia do usuário do AWS Systems Manager.

Exemplo 3: Para obter uma lista de patches instalados desde um especificado InstalledTime para uma instância

O exemplo de `describe-instance-patches` a seguir recupera informações sobre patches instalados desde um horário especificado para a instância especificada combinando o uso de `--filters` e `--query`.

```
aws ssm describe-instance-patches \  
  --instance-id "i-1234567890abcdef0" \  
  --filters Key=State,Values=Installed \  
  --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

Saída:

```
{
  "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--"
}
```

- Para API obter detalhes, consulte [DescribeInstancePatches](#) na Referência de AWS CLI Comandos.

describe-inventory-deletions

O código de exemplo a seguir mostra como usar `describe-inventory-deletions`.

AWS CLI

Para obter exclusões de inventário

Este exemplo recupera detalhes das operações de exclusão de inventário.

Comando:

```
aws ssm describe-inventory-deletions
```

Saída:

```
{
  "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
  }
]
}

```

Para obter detalhes de uma exclusão específica de inventário

Este exemplo recupera detalhes de uma operação específica de exclusão de inventário.

Comando:

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-b1c6-987654321c4d"
```

Saída:

```
{
  "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
    }
  ]
}
```

- Para API obter detalhes, consulte [DescribeInventoryDeletions](#) na Referência de AWS CLI Comandos.

describe-maintenance-window-execution-task-invocations

O código de exemplo a seguir mostra como usar `describe-maintenance-window-execution-task-invocations`.

AWS CLI

Para fazer com que as invocações da tarefa específica sejam realizadas para a execução de tarefa de uma janela de manutenção

O exemplo de `describe-maintenance-window-execution-task-invocations` a seguir lista as invocações para a tarefa especificada executada como parte da execução da janela de manutenção especificada.

```
aws ssm describe-maintenance-window-execution-task-invocations \
```

```
--window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \  
--task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Exibir informações sobre tarefas e execuções de tarefas \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowExecutionTasks](#) na Referência de AWS CLI Comandos.

describe-maintenance-window-execution-tasks

O código de exemplo a seguir mostra como usar `describe-maintenance-window-execution-tasks`.

AWS CLI

Para listar todas as tarefas associadas à execução de uma janela de manutenção

O exemplo de `ssm describe-maintenance-window-execution-tasks` a seguir lista as tarefas associadas à execução da janela de manutenção especificada.

```
aws ssm describe-maintenance-window-execution-tasks \  
--window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Exibir informações sobre tarefas e execuções de tarefas \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowExecutions](#) na Referência de AWS CLI Comandos.

describe-maintenance-window-executions

O código de exemplo a seguir mostra como usar `describe-maintenance-window-executions`.

AWS CLI

Exemplo 1: listar todas as execuções para uma janela de manutenção

O exemplo de `describe-maintenance-window-executions` a seguir lista todas as execuções da janela de manutenção especificada.

```
aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE"
```

Saída:

```
{
  "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"
  }
]
}

```

Exemplo 2: listar todas as execuções para uma janela de manutenção antes de uma data especificada

O exemplo de `describe-maintenance-window-executions` a seguir lista todas as execuções da janela de manutenção especificada antes da data especificada.

```

aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"

```

Saída:

```

{
  "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"
  }
]
}

```

Exemplo 3: listar todas as execuções para uma janela de manutenção após uma data especificada

O exemplo de `describe-maintenance-window-executions` a seguir lista todas as execuções da janela de manutenção especificada após a data especificada.

```

aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"

```

Saída:

```

{
  "WindowExecutions": [
    {
      "WindowId": "mw-ab12cd34eEXAMPLE",
      "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
      "Status": "IN_PROGRESS",
      "StartTime": "2021-08-04T11:00:00.000000-07:00"
    }
  ]
}

```

Para obter mais informações, consulte [Exibir informações sobre tarefas e execuções de tarefas \(AWS CLI\)](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowExecutions](#) na Referência de AWS CLI Comandos.

describe-maintenance-window-schedule

O código de exemplo a seguir mostra como usar `describe-maintenance-window-schedule`.

AWS CLI

Exemplo 1: Para listar as próximas execuções para uma janela de manutenção

O `describe-maintenance-window-schedule` exemplo a seguir lista todas as próximas execuções para a janela de manutenção especificada.

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-ab12cd34eEXAMPLE
```

Saída:

```
{  
  "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"  
    },  
    ...  
  ]  
}
```

Exemplo 2: Para listar todas as próximas execuções de uma janela de manutenção antes de uma data especificada

O `describe-maintenance-window-schedule` exemplo a seguir lista todas as execuções futuras da janela de manutenção especificada que ocorram antes da data especificada.

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-0ecb1226dd7b2e9a6 \  
  --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

Exemplo 3: Para listar todas as próximas execuções de uma janela de manutenção após uma data especificada

O `describe-maintenance-window-schedule` exemplo a seguir lista todas as execuções futuras da janela de manutenção especificada que ocorram após a data especificada.

```
aws ssm describe-maintenance-window-schedule \  
  --window-id mw-0ecb1226dd7b2e9a6 \  
  --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

```
--window-id mw-0ecb1226dd7b2e9a6 \  
--filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowSchedule](#) na Referência de AWS CLI Comandos.

describe-maintenance-window-targets

O código de exemplo a seguir mostra como usar `describe-maintenance-window-targets`.

AWS CLI

Exemplo 1: listar todos os destinos para uma janela de manutenção

O exemplo de `describe-maintenance-window-targets` a seguir lista todos os destinos para uma janela de manutenção.

```
aws ssm describe-maintenance-window-targets \  
--window-id "mw-06cf17cbefEXAMPLE"
```

Saída:

```
{  
  "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"
  }
]
}

```

Exemplo 2: listar todos os destinos para uma janela de manutenção que correspondem ao valor das informações de um proprietário específico

Esse exemplo de `describe-maintenance-window-targets` lista todos os destinos de uma janela de manutenção com um valor específico.

```

aws ssm describe-maintenance-window-targets \
  --window-id "mw-0ecb1226ddEXAMPLE" \
  --filters "Key=OwnerInformation,Values=CostCenter1"

```

Saída:

```

{
  "Targets": [
    {
      "WindowId": "mw-0ecb1226ddEXAMPLE",
      "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
      "ResourceType": "INSTANCE",
      "Targets": [
        {
          "Key": "tag:Environment",
          "Values": [
            "Prod"
          ]
        }
      ]
    },
    "OwnerInformation": "CostCenter1",
  ]
}

```

```

        "Name": "ProdTarget1"
      }
    ]
  }

```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowTargets](#) na Referência de AWS CLI Comandos.

describe-maintenance-window-tasks

O código de exemplo a seguir mostra como usar `describe-maintenance-window-tasks`.

AWS CLI

Exemplo 1: listar todas as tarefas para uma janela de manutenção

O exemplo de `describe-maintenance-window-tasks` a seguir lista todas as tarefas para a janela de manutenção especificada.

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-06cf17cbefEXAMPLE"

```

Saída:

```

{
  "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"
            ]
        }
    ]
}
]
}
}

```

Exemplo 2: Para listar todas as tarefas de uma janela de manutenção que invoca o documento de comando AWS- RunPowerShellScript

O exemplo de `describe-maintenance-window-tasks` a seguir lista todas as tarefas para a janela de manutenção especificada que invoca o documento de comandos do AWS- RunPowerShellScript.

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \

```

```
--filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 3: listar todas as tarefas de uma janela de manutenção que têm a prioridade igual a 3

O exemplo de `describe-maintenance-window-tasks` a seguir lista todas as tarefas para a janela de manutenção especificada que tem `Priority` igual a 3.

```
aws ssm describe-maintenance-window-tasks \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=Priority,Values=3"
```

Saída:

```
{
```

```
"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"
  }
]
```

```
]
}
```

Exemplo 4: listar todas as tarefas de uma janela de manutenção que têm a prioridade igual a 1 e usam o Run Command

Esse exemplo de `describe-maintenance-window-tasks` lista todas as tarefas para a janela de manutenção especificada que tem `Priority` igual a 1 e usam `Run Command`.

```
aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowTasks](#) na Referência de AWS CLI Comandos.

describe-maintenance-windows-for-target

O código de exemplo a seguir mostra como usar `describe-maintenance-windows-for-target`.

AWS CLI

Para listar todas as janelas de manutenção associadas a uma instância específica

O `describe-maintenance-windows-for-target` exemplo a seguir lista as janelas de manutenção que têm alvos ou tarefas associadas à instância especificada.

```
aws ssm describe-maintenance-windows-for-target \  
  --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \  
  --resource-type INSTANCE
```

Saída:

```
{  
  "WindowIdentities": [  
    {  
      "WindowId": "mw-0c5ed765acEXAMPLE",  
      "Name": "My-First-Maintenance-Window"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindowsForTarget](#) na Referência de AWS CLI Comandos.

describe-maintenance-windows

O código de exemplo a seguir mostra como usar `describe-maintenance-windows`.

AWS CLI

Exemplo 1: listar todas as janelas de manutenção

O `describe-maintenance-windows` exemplo a seguir lista todas as janelas de manutenção em sua AWS conta na região atual.

```
aws ssm describe-maintenance-windows
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: listar todas as janelas de manutenção habilitadas

O exemplo de `describe-maintenance-windows` a seguir lista todas as janelas de manutenção habilitadas.

```
aws ssm describe-maintenance-windows \
  --filters "Key=Enabled,Values=true"
```


Exemplo 3: listar janelas de manutenção que correspondem a um nome específico

Esse exemplo de `describe-maintenance-windows` lista todas as janelas de manutenção com o nome especificado.

```
aws ssm describe-maintenance-windows \  
  --filters "Key=Name,Values=MyMaintenanceWindow"
```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeMaintenanceWindows](#) na Referência de AWS CLI Comandos.

describe-ops-items

O código de exemplo a seguir mostra como usar `describe-ops-items`.

AWS CLI

Para listar um conjunto de OpsItems

O `describe-ops-items` exemplo a seguir exibe uma lista de todos os abertos OpsItems em sua AWS conta.

```
aws ssm describe-ops-items \  
  --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Saída:

```
{  
  "OpsItemSummaries": [  
    {  
      "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/  
fbf77cbe264a33509569f23e4EXAMPLE",  
      "CreatedTime": "2020-03-14T17:02:46.375000-07:00",  
      "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-  
Role/fbf77cbe264a33509569f23e4EXAMPLE",  
      "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",  
      "Source": "SSM",  
      "Status": "Open",  
      "OpsItemId": "oi-7cfc5EXAMPLE",
```

```

    "Title": "SSM Maintenance Window execution failed",
    "OperationalData": {
      "/aws/dedup": {
        "Value": "{\"dedupString\": \"SSMOpsItems-SSM-maintenance-window-
execution-failed\"}",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[{\"arn\": \"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
        "Type": "SearchableString"
      }
    },
    "Category": "Availability",
    "Severity": "3"
  },
  {
    "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
    "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
    "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
    "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
    "Source": "EC2",
    "Status": "Open",
    "OpsItemId": "oi-6f966EXAMPLE",
    "Title": "EC2 instance stopped",
    "OperationalData": {
      "/aws/automations": {
        "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
\"automationId\": \"AWS-RestartEC2Instance\" } ]",
        "Type": "SearchableString"
      },
      "/aws/dedup": {
        "Value": "{\"dedupString\": \"SSMOpsItems-EC2-instance-stopped
\"}",
        "Type": "SearchableString"
      },
      "/aws/resources": {
        "Value": "[{\"arn\": \"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
        "Type": "SearchableString"
      }
    }
  },

```

```

        "Category": "Availability",
        "Severity": "3"
    }
]
}

```

Para obter mais informações, consulte [Trabalhando com OpsItems](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeOpsItems](#) na Referência de AWS CLI Comandos.

describe-parameters

O código de exemplo a seguir mostra como usar `describe-parameters`.

AWS CLI

Exemplo 1: como listar todos os parâmetros

O `describe-parameters` exemplo a seguir lista todos os parâmetros na AWS conta atual e na região.

```
aws ssm describe-parameters
```

Saída:

```

{
  "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": []
  }
]
}

```

Exemplo 2: como listar todos os parâmetros que correspondem a metadados específicos

Esse exemplo de `describe-parameters` lista todos os parâmetros que correspondem a um filtro.

```
aws ssm describe-parameters --filters "Chave = Tipo, Valores =" StringList
```

Saída:

```
{
  "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": []
    }
  ]
}
```

Para obter mais informações, consulte [Pesquisando parâmetros do Systems Manager](#), no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeParameters](#) na Referência de AWS CLI Comandos.

describe-patch-baselines

O código de exemplo a seguir mostra como usar `describe-patch-baselines`.

AWS CLI

Exemplo 1: Para listar todas as linhas de base do patch

O exemplo de `describe-patch-baselines` a seguir recupera os detalhes de todas as listas de referências de patches da sua conta na região atual.

```
aws ssm describe-patch-baselines
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: Para listar todas as linhas de base do patch fornecidas pelo AWS

O `describe-patch-baselines` exemplo a seguir lista todas as linhas de base de patch fornecidas pelo AWS

```
aws ssm describe-patch-baselines \
  --filters "Key=OWNER,Values=[AWS]"
```

Exemplo 3: listar todas as listas de referência de patches pertencentes a você

O exemplo de `describe-patch-baselines` a seguir lista todas as listas de referências de patches criadas em sua conta na região atual.

```
aws ssm describe-patch-baselines \  
  --filters "Key=OWNER,Values=[Self]"
```

Para obter mais informações, consulte [Sobre listas de referência de patches predefinidas e personalizadas](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribePatchBaselines](#) na Referência de AWS CLI Comandos.

`describe-patch-group-state`

O código de exemplo a seguir mostra como usar `describe-patch-group-state`.

AWS CLI

Para obter o estado de um grupo de patches

O exemplo de `describe-patch-group-state` a seguir recupera o resumo de conformidade de patches de alto nível para um grupo de patches.

```
aws ssm describe-patch-group-state \  
  --patch-group "Production"
```

Saída:

```
{  
  "Instances": 21,  
  "InstancesWithCriticalNonCompliantPatches": 1,  
  "InstancesWithFailedPatches": 2,  
  "InstancesWithInstalledOtherPatches": 3,  
  "InstancesWithInstalledPatches": 21,  
  "InstancesWithInstalledPendingRebootPatches": 2,  
  "InstancesWithInstalledRejectedPatches": 1,  
  "InstancesWithMissingPatches": 3,  
  "InstancesWithNotApplicablePatches": 4,  
  "InstancesWithOtherNonCompliantPatches": 1,  
  "InstancesWithSecurityNonCompliantPatches": 1,  
  "InstancesWithUnreportedNotApplicablePatches": 2
```

```
}
```

Para obter mais informações, consulte Sobre grupos de patches < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html>>__ e Entendendo os valores do [estado de conformidade](#) de patches no Guia do Usuário do Systems Manager.AWS

- Para API obter detalhes, consulte [DescribePatchGroupState](#) na Referência de AWS CLI Comandos.

describe-patch-groups

O código de exemplo a seguir mostra como usar describe-patch-groups.

AWS CLI

Para exibir registros de grupos de patches

O exemplo de describe-patch-groups a seguir lista os registros de grupos de patches.

```
aws ssm describe-patch-groups
```

Saída:

```
{
  "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
      }
    },
    ...
  ]
}

```

Para obter mais informações, consulte Criar um grupo de patches < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> > e [Adicionar um grupo de patches a uma linha de base de patches no Guia do usuário do Systems Manager.AWS](#)

- Para API obter detalhes, consulte [DescribePatchGroups](#) na Referência de AWS CLI Comandos.

describe-patch-properties

O código de exemplo a seguir mostra como usar describe-patch-properties.

AWS CLI

Para listar a disponibilidade de patches do Amazon Linux

O describe-patch-properties exemplo a seguir exibe uma lista dos produtos Amazon Linux para os quais os patches estão disponíveis em sua AWS conta.

```

aws ssm describe-patch-properties \
  --operating-system AMAZON_LINUX \
  --property PRODUCT

```

Saída:

```

{
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Sobre listas de referência de patches](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribePatchProperties](#) na Referência de AWS CLI Comandos.

describe-sessions

O código de exemplo a seguir mostra como usar `describe-sessions`.

AWS CLI

Exemplo 1: Para listar todas as sessões ativas do Gerenciador de Sessões

Este `describe-sessions` exemplo recupera uma lista das sessões ativas criadas mais recentemente (sessões conectadas e desconectadas) nos últimos 30 dias que foram iniciadas pelo usuário especificado. Esse comando retorna somente resultados para conexões com destinos iniciadas usando o Gerenciador de Sessões. Ele não lista as conexões feitas por outros meios, como conexões de área de trabalho remota ou SSH.

```
aws ssm describe-sessions \  
  --state "Active" \  
  --filters "key=Owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/  
Shirley-Rodriguez"
```

Saída:

```
{  
  "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": {}  
    }  
  ]  
}
```

Exemplo 2: Para listar todas as sessões encerradas do Gerenciador de Sessões

Este `describe-sessions` exemplo recupera uma lista das sessões encerradas mais recentemente dos últimos 30 dias para todos os usuários.

```
aws ssm describe-sessions \  
  --state "History"
```

Saída:

```
{  
  "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--"  
}
```

Para obter mais informações, consulte [Exibir histórico da sessão](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [DescribeSessions](#) na Referência de AWS CLI Comandos.

disassociate-ops-item-related-item

O código de exemplo a seguir mostra como usar `disassociate-ops-item-related-item`.

AWS CLI

Para excluir uma associação de item relacionado

O `disassociate-ops-item-related-item` exemplo a seguir exclui a associação entre o `OpsItem` e um item relacionado.

```
aws ssm disassociate-ops-item-related-item \  
  --ops-item-id "oi-f99f2EXAMPLE" \  
  --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Trabalhando com incidentes do Incident Manager OpsCenter no Guia do Usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [DisassociateOpsItemRelatedItem](#) na Referência de AWS CLI Comandos.

`get-automation-execution`

O código de exemplo a seguir mostra como usar `get-automation-execution`.

AWS CLI

Para exibir detalhes sobre uma execução do Automation

O exemplo de `get-automation-execution` a seguir exibe informações detalhadas sobre uma execução do Automation.

```
aws ssm get-automation-execution \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Saída:

```
{  
  "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
    }
  }
}

```

Para obter mais informações, consulte [Passo a passo: Patch a Linux AMI \(AWS CLI\) no Guia do Usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [GetAutomationExecution](#) na Referência de AWS CLI Comandos.

get-calendar-state

O código de exemplo a seguir mostra como usar `get-calendar-state`.

AWS CLI

Exemplo 1: Para obter o estado atual de um calendário de alterações

Este `get-calendar-state` exemplo retorna o estado de um calendário no horário atual. Como o exemplo não especifica um horário, o estado atual do calendário é relatado.

```
aws ssm get-calendar-state \  
  --calendar-names "MyCalendar"
```

Saída:

```
{  
  "State": "OPEN",  
  "AtTime": "2020-02-19T22:28:51Z",  
  "NextTransitionTime": "2020-02-24T21:15:19Z"  
}
```

Exemplo 2: Para obter o estado de um calendário de alterações em um horário especificado

Este `get-calendar-state` exemplo retorna o estado de um calendário no horário especificado.

```
aws ssm get-calendar-state \  
  --calendar-names "MyCalendar" \  
  --at-time "2020-07-19T21:15:19Z"
```

Saída:

```
{  
  "State": "CLOSED",  
  "AtTime": "2020-07-19T21:15:19Z"  
}
```

Para obter mais informações, consulte [Get the State of the Change Calendar](#) no AWS Systems Manager User Guide.

- Para API obter detalhes, consulte [GetCalendarState](#) na Referência de AWS CLI Comandos.

get-command-invocation

O código de exemplo a seguir mostra como usar `get-command-invocation`.

AWS CLI

Para exibir os detalhes de uma invocação de comando

O exemplo de `get-command-invocation` a seguir lista todas as invocações do comando especificado na instância especificada.

```
aws ssm get-command-invocation \  
  --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \  
  --instance-id "i-1234567890abcdef0"
```

Saída:

```
{  
  "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",  
  "InstanceId": "i-1234567890abcdef0",  
  "Comment": "b48291dd-ba76-43e0-b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",  
  "DocumentName": "AWS-UpdateSSMAgent",  
  "DocumentVersion": "",  
  "PluginName": "aws:updateSsmAgent",  
  "ResponseCode": 0,  
  "ExecutionStartDateTime": "2020-02-19T18:18:03.419Z",  
  "ExecutionElapsedTime": "PT0.091S",  
  "ExecutionEndDateTime": "2020-02-19T18:18:03.419Z",  
  "Status": "Success",  
  "StatusDetails": "Success",  
  "StandardOutputContent": "Updating amazon-ssm-agent from 2.3.842.0 to latest  
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/  
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,  
update skipped\n",  
  "StandardOutputUrl": "",  
  "StandardErrorContent": "",  
  "StandardErrorUrl": "",
```



```
"CloudWatchOutputConfig": {
  "CloudWatchLogGroupName": "",
  "CloudWatchOutputEnabled": false
}
```

Para obter mais informações, consulte [Entender os status dos comandos](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetCommandInvocation](#) na Referência de AWS CLI Comandos.

get-connection-status

O código de exemplo a seguir mostra como usar `get-connection-status`.

AWS CLI

Para exibir o status da conexão de uma instância gerenciada

Este exemplo de `get-connection-status` retorna o status da conexão da instância gerenciada especificada.

```
aws ssm get-connection-status \
  --target i-1234567890abcdef0
```

Saída:

```
{
  "Target": "i-1234567890abcdef0",
  "Status": "connected"
}
```

- Para API obter detalhes, consulte [GetConnectionStatus](#) na Referência de AWS CLI Comandos.

get-default-patch-baseline

O código de exemplo a seguir mostra como usar `get-default-patch-baseline`.

AWS CLI

Exemplo 1: para exibir a lista de referência de patches padrão do Windows

O exemplo de `get-default-patch-baseline` a seguir recupera detalhes da lista de referência de patches padrão para o Windows Server.

```
aws ssm get-default-patch-baseline
```

Saída:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Exemplo 2: para exibir a lista de referência de patches padrão do Amazon Linux

O exemplo de `get-default-patch-baseline` a seguir recupera detalhes da lista de referência de patches padrão para o Amazon Linux.

```
aws ssm get-default-patch-baseline \
  --operating-system AMAZON_LINUX
```

Saída:

```
{
  "BaselineId": "pb-047c6eb9c8fc12345",
  "OperatingSystem": "AMAZON_LINUX"
}
```

Para obter mais informações, consulte [Sobre linhas de base de patches predefinidas e personalizadas < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html) e Definir uma linha de base de [patch existente](#) como padrão no Guia do usuário do Systems Manager.AWS

- Para API obter detalhes, consulte [GetDefaultPatchBaseline](#) na Referência de AWS CLI Comandos.

get-deployable-patch-snapshot-for-instance

O código de exemplo a seguir mostra como usar `get-deployable-patch-snapshot-for-instance`.

AWS CLI

Para recuperar o instantâneo atual da lista de referência de patches usado por uma instância

O exemplo de `get-deployable-patch-snapshot-for-instance` a seguir recupera detalhes do instantâneo atual da lista de referência de patches especificada usada por uma instância. Esse comando deve ser executado da instância usando as credenciais da instância. Para garantir que ele use as credenciais da instância, execute `aws configure` e especifique somente a região da sua instância. Deixe os campos `Access Key` e `Secret Key` vazios.

Dica: use `uuidgen` para gerar um `snapshot-id`.

```
aws ssm get-deployable-patch-snapshot-for-instance \
  --instance-id "i-1234567890abcdef0" \
  --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Saída:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "SnapshotId": "521c3536-930c-4aa9-950e-01234567abcd",
  "Product": "AmazonLinux2018.03",
  "SnapshotDownloadUrl": "https://patch-baseline-snapshot-us-east-1.s3.amazonaws.com/ed85194ef27214f5984f28b4d664d14f7313568fea7d4b6ac6c10ad1f729d7e7-773304212436/AMAZON_LINUX-521c3536-930c-4aa9-950e-01234567abcd?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190215T164031Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=AKIAJ5C56P35AEBRX2QQ%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
}
```

Para obter mais informações, consulte [Nome do parâmetro: ID do instantâneo](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetDeployablePatchSnapshotForInstance](#) na Referência de AWS CLI Comandos.

get-document

O código de exemplo a seguir mostra como usar `get-document`.

AWS CLI

Para obter conteúdo do documento

O exemplo de `get-document` a seguir exibe o conteúdo de um documento do Systems Manager.

```
aws ssm get-document \
  --name "AWS-RunShellScript"
```

Saída:

```
{
  "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\": \"(Required) Specify a shell script or a command to run.\",\n      \"minItems\": 1,\n      \"displayType\": \"textarea\",\n      \"workingDirectory\": {\n        \"type\": \"String\",\n        \"default\": \"\",\n        \"description\": \"(Optional) The path to the working\n        directory on your instance.\",\n        \"maxChars\": 4096\n      },\n      \"executionTimeout\": {\n        \"type\": \"String\",\n        \"default\": \"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][0-9]{0,4})|(1[0-6][0-9]{4})|(17[0-1][0-9]{3})|(172[0-7][0-9]{2})|(172800)\"\n      }\n    },\n    \"runtimeConfig\": {\n      \"aws:runShellScript\": {\n        \"properties\": [\n          {\n            \"id\": \"0.aws:runShellScript\",\n            \"runCommand\": \"{{ commands }}\",\n            \"workingDirectory\": \"{{ workingDirectory }}\",\n            \"timeoutSeconds\": \"{{ executionTimeout }}\"\n          }\n        ]\n      }\n    }\n  },\n  \"DocumentType\": \"Command\",\n  \"DocumentFormat\": \"JSON\"\n}
```

Para obter mais informações, consulte [Documentos do AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetDocument](#) na Referência de AWS CLI Comandos.

get-inventory-schema

O código de exemplo a seguir mostra como usar `get-inventory-schema`.

AWS CLI

Para visualizar seu esquema de inventário

Este exemplo retorna uma lista de nomes de tipos de inventário para a conta.

Comando:

```
aws ssm get-inventory-schema
```

Saída:

```
{
  "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--"
}

```

Para visualizar o esquema de inventário de um tipo de inventário específico

Este exemplo retorna o esquema de inventário para o tipo de inventário AWS:AWS Componente.

Comando:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- Para API obter detalhes, consulte [GetInventorySchema](#) na Referência de AWS CLI Comandos.

get-inventory

O código de exemplo a seguir mostra como usar `get-inventory`.

AWS CLI

Para visualizar o inventário

Este exemplo obtém os metadados personalizados do seu inventário.

Comando:

```
aws ssm get-inventory
```

Saída:

```

{
  "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"
  }
]
}

```

- Para API obter detalhes, consulte [GetInventory](#) na Referência de AWS CLI Comandos.

get-maintenance-window-execution-task-invocation

O código de exemplo a seguir mostra como usar `get-maintenance-window-execution-task-invocation`.

AWS CLI

Para obter informações sobre a invocação de uma tarefa da janela de manutenção

O `get-maintenance-window-execution-task-invocation` exemplo a seguir lista informações sobre a chamada de tarefa especificada que faz parte da execução da janela de manutenção especificada.

```
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"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Exibir informações sobre tarefas e execuções de tarefas \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetMaintenanceWindowExecutionTaskInvocation](#) na Referência de AWS CLI Comandos.

get-maintenance-window-execution-task

O código de exemplo a seguir mostra como usar `get-maintenance-window-execution-task`.

AWS CLI

Para obter informações sobre a execução de uma tarefa da janela de manutenção

O exemplo de `get-maintenance-window-execution-task` a seguir lista informações sobre uma tarefa que faz parte da execução da janela de manutenção especificada.

```
aws ssm get-maintenance-window-execution-task \  
--window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \  
--task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```


Saída:

```
{
  "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"  
}
```

Para obter mais informações, consulte [Exibir informações sobre tarefas e execuções de tarefas \(AWS CLI\)](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetMaintenanceWindowExecutionTask](#) na Referência de AWS CLI Comandos.

get-maintenance-window-execution

O código de exemplo a seguir mostra como usar `get-maintenance-window-execution`.

AWS CLI

Para obter informações sobre a execução de uma tarefa da janela de manutenção

O exemplo de `get-maintenance-window-execution` a seguir lista informações sobre uma tarefa que é executada como parte da execução da janela de manutenção especificada.

```
aws ssm get-maintenance-window-execution \  
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Saída:

```
{  
  "Status": "SUCCESS",  
  "TaskIds": [  
    "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"  
  ],  
  "StartTime": 1487692834.595,  
  "EndTime": 1487692835.051,  
  "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",  
}
```

Para obter mais informações, consulte [Exibir informações sobre tarefas e execuções de tarefas \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetMaintenanceWindowExecution](#) na Referência de AWS CLI Comandos.

get-maintenance-window-task

O código de exemplo a seguir mostra como usar `get-maintenance-window-task`.

AWS CLI

Para obter informações sobre uma tarefa da janela de manutenção

O `get-maintenance-window-task` exemplo a seguir recupera detalhes sobre a tarefa especificada da janela de manutenção.

```
aws ssm get-maintenance-window-task \  
  --window-id mw-0c5ed765acEXAMPLE \  
  --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

Saída:

```
{  
  "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"
}

```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetMaintenanceWindowTask](#) na Referência de AWS CLI Comandos.

get-maintenance-window

O código de exemplo a seguir mostra como usar `get-maintenance-window`.

AWS CLI

Para obter informações sobre uma janela de manutenção

O exemplo de `get-maintenance-window` a seguir recupera detalhes sobre a janela de manutenção especificada.

```
aws ssm get-maintenance-window \
  --window-id "mw-03eb9db428EXAMPLE"
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Exibir informações sobre janelas de manutenção \(AWS CLI\)](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetMaintenanceWindow](#) na Referência de AWS CLI Comandos.

get-ops-item

O código de exemplo a seguir mostra como usar `get-ops-item`.

AWS CLI

Para ver informações sobre um OpsItem

O `get-ops-item` exemplo a seguir exibe detalhes sobre o especificado OpsItem.

```
aws ssm get-ops-item \
  --ops-item-id oi-0b725EXAMPLE
```

Saída:

```
{
  "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"
  }
}

```

Para obter mais informações, consulte [Trabalhando com OpsItems](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetOpsItem](#) na Referência de AWS CLI Comandos.

get-ops-summary

O código de exemplo a seguir mostra como usar `get-ops-summary`.

AWS CLI

Para ver um resumo de todos OpsItems

O `get-ops-summary` exemplo a seguir exibe um resumo de tudo OpsItems em sua AWS conta.

```
aws ssm get-ops-summary
```

Saída:

```
{
  "Entities": [
    {
      "Id": "oi-4309fEXAMPLE",
      "Data": {
        "AWS:OpsItem": {
          "CaptureTime": "2020-02-26T18:58:32.918Z",
          "Content": [
            {
              "AccountId": "111222333444",
              "Category": "Availability",
              "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
              "CreatedTime": "2020-02-26T19:10:44.149Z",
              "Description": "CloudWatch Event Rule SSMOpsItems-EC2-
instance-terminated was triggered. Your EC2 instance has terminated. See below for
more details.",
              "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
              "LastModifiedTime": "2020-02-26T19:10:44.149Z",
              "Notifications": "",
              "OperationalData": "{\"/aws/automations\":
{\"type\": \"SearchableString\", \"value\": \"[ { \\\"automationType\\\": \\
\\\"AWS:SSM:Automation\\\"\", \\\"automationId\\\": \\\"AWS-CreateManagedWindowsInstance
\\\" } ], { \\\"automationType\\\": \\\"AWS:SSM:Automation\\\"\", \\\"automationId
\\\": \\\"AWS-CreateManagedLinuxInstance\\\" } ]\"}, \"/aws/resources\":
{\"type\": \"SearchableString\", \"value\": \"[{\\\"arn\\\": \\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0acbd0800fEXAMPLE\\\"}]\"}, \"/aws/dedup\": {\"type\":
\\\"SearchableString\", \"value\": \"{\\\"dedupString\\\": \\\"SSMOpsItems-EC2-instance-
terminated\\\"}\"}}",
            }
          ]
        }
      }
    }
  ]
}
```

```

        "OpsItemId": "oi-4309fEXAMPLE",
        "RelatedItems": "",
        "Severity": "3",
        "Source": "EC2",
        "Status": "Open",
        "Title": "EC2 instance terminated"
    }
  ]
}
},
{
  "Id": "oi-bb2a0e6a4541",
  "Data": {
    "AWS:OpsItem": {
      "CaptureTime": "2019-11-26T19:20:06.161Z",
      "Content": [
        {
          "AccountId": "111222333444",
          "Category": "Availability",
          "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
          "CreatedTime": "2019-11-26T20:00:07.237Z",
          "Description": "CloudWatch Event Rule SSM0psItems-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\\\": \\\"SSM0psItems-SSM-
maintenance-window-execution-failed\\\"}\"}}",
          "OpsItemId": "oi-bb2a0EXAMPLE",
          "RelatedItems": "",
          "Severity": "3",
          "Source": "SSM",
          "Status": "Open",
          "Title": "SSM Maintenance Window execution failed"
        }
      ]
    }
  }
}

```



```
    }  
  }  
]  
}
```

Para obter mais informações, consulte [Trabalhando com OpsItems](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetOpsSummary](#) na Referência de AWS CLI Comandos.

get-parameter-history

O código de exemplo a seguir mostra como usar `get-parameter-history`.

AWS CLI

Para obter o histórico de valores de um parâmetro

O exemplo de `get-parameter-history` a seguir lista o histórico de alterações do parâmetro especificado, incluindo seu valor.

```
aws ssm get-parameter-history \  
  --name "MyStringParameter"
```

Saída:

```
{  
  "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": []
    }
]
}

```

Para obter mais informações, consulte [Trabalhar com versões de parâmetros](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetParameterHistory](#) na Referência de AWS CLI Comandos.

get-parameter

O código de exemplo a seguir mostra como usar `get-parameter`.

AWS CLI

Exemplo 1: exibir o valor de um parâmetro

O exemplo de `get-parameter` a seguir lista o valor do parâmetro único especificado.

```
aws ssm get-parameter \
  --name "MyStringParameter"
```

Saída:

```
{
  "Parameter": {
    "Name": "MyStringParameter",
    "Type": "String",
    "Value": "Veni",
    "Version": 1,
    "LastModifiedDate": 1530018761.888,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
    "DataType": "text"
  }
}
```

Para obter mais informações, consulte [Trabalhar com o Parameter Store](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: Para descriptografar o valor de um parâmetro SecureString

O exemplo de `get-parameter` a seguir descriptografa o valor do parâmetro SecureString especificado.

```
aws ssm get-parameter \
  --name "MySecureStringParameter" \
  --with-decryption
```

Saída:

```
{
  "Parameter": {
    "Name": "MySecureStringParameter",
    "Type": "SecureString",
    "Value": "16679b88-310b-4895-a943-e0764EXAMPLE",
    "Version": 2,
    "LastModifiedDate": 1582155479.205,
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MySecureStringParameter"
    "DataType": "text"
  }
}
```

Para obter mais informações, consulte [Trabalhar com o Parameter Store](#) no Guia do usuário do AWS Systems Manager.

Exemplo 3: exibir o valor de um parâmetro usando rótulos

O exemplo de `get-parameter` a seguir lista o valores do parâmetros especificado com um rótulo especificado.

```
aws ssm get-parameter \  
  --name "MyParameter:label"
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Trabalhar com rótulos de parâmetros](#) no Guia do usuário do AWS Systems Manager.

Exemplo 4: exibir o valor de um parâmetro usando versões

O exemplo de `get-parameter` a seguir lista o valor da versão do parâmetro único especificado.

```
aws ssm get-parameter \  
  --name "MyParameter:2"
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Trabalhar com rótulos de parâmetros](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetParameter](#) na Referência de AWS CLI Comandos.

get-parameters-by-path

O código de exemplo a seguir mostra como usar `get-parameters-by-path`.

AWS CLI

Para listar parâmetros em um caminho específico

O `get-parameters-by-path` exemplo a seguir lista os parâmetros dentro da hierarquia especificada.

```
aws ssm get-parameters-by-path \  
  --path "/site/newyork/department/"
```

Saída:

```
{  
  "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"
    },
    ...
]
}

```

Para obter mais informações, consulte [Trabalhando com hierarquias de parâmetros no Guia do Usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [GetParametersByPath](#) na Referência de AWS CLI Comandos.

get-parameters

O código de exemplo a seguir mostra como usar `get-parameters`.

AWS CLI

Exemplo 1: listar os valores de um parâmetro

O exemplo de `get-parameters` a seguir lista os valores dos três parâmetros especificados.

```

aws ssm get-parameters \
  --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"

```

Saída:

```

{
  "Parameters": [
    {
      "Name": "MyStringListParameter",
      "Type": "StringList",
      "Value": "alpha,beta,gamma",
      "Version": 1,
      "LastModifiedDate": 1582154764.222,
      "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/
MyStringListParameter"
    }
  ]
}

```

```

        "DataType": "text"
    },
    {
        "Name": "MyStringParameter",
        "Type": "String",
        "Value": "Vici",
        "Version": 3,
        "LastModifiedDate": 1582156117.545,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
],
"InvalidParameters": [
    "MyInvalidParameterName"
]
}

```

Para obter mais informações, consulte [Trabalhar com o Parameter Store](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: listar nomes e valores de vários parâmetros usando a opção "--query"

O exemplo de `get-parameters` a seguir lista os nomes e valores dos parâmetros especificados.

```

aws ssm get-parameters \
  --names MyStringParameter MyStringListParameter \
  --query "Parameters[*].{Name:Name, Value:Value}"

```

Saída:

```

[
  {
    "Name": "MyStringListParameter",
    "Value": "alpha,beta,gamma"
  },
  {
    "Name": "MyStringParameter",
    "Value": "Vidi"
  }
]

```

Para obter mais informações, consulte [Trabalhar com o Parameter Store](#) no Guia do usuário do AWS Systems Manager.

Exemplo 3: exibir o valor de um parâmetro usando rótulos

O exemplo de `get-parameter` a seguir lista o valores do parâmetros especificado com um rótulo especificado.

```
aws ssm get-parameter \  
  --name "MyParameter:Label"
```

Saída:

```
{  
  "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": []  
}
```

Para obter mais informações, consulte [Trabalhar com rótulos de parâmetros](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetParameters](#) na Referência de AWS CLI Comandos.

get-patch-baseline-for-patch-group

O código de exemplo a seguir mostra como usar `get-patch-baseline-for-patch-group`.

AWS CLI

Para exibir uma lista de referência de patches de um grupo de patches

O exemplo de `get-patch-baseline-for-patch-group` a seguir recupera detalhes sobre a lista de referência de patches para a instância especificada.

```
aws ssm get-patch-baseline-for-patch-group \  
  --patch-group "DEV"
```

Saída:

```
{  
  "PatchGroup": "DEV",  
  "BaselineId": "pb-0123456789abcdef0",  
  "OperatingSystem": "WINDOWS"  
}
```

Para obter mais informações, consulte Criar um grupo de patches < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> > e [Adicionar um grupo de patches a uma linha de base de patches no Guia do usuário do Systems Manager](#).AWS

- Para API obter detalhes, consulte [GetPatchBaselineForPatchGroup](#) na Referência de AWS CLI Comandos.

get-patch-baseline

O código de exemplo a seguir mostra como usar `get-patch-baseline`.

AWS CLI

Para exibir uma lista de referência de patches

O exemplo de `get-patch-baseline` a seguir recupera os detalhes da lista de referência de patches especificada.

```
aws ssm get-patch-baseline \  
--baseline-id "pb-0123456789abcdef0"
```

Saída:

```
{  
  "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": []  
}
```

Para obter mais informações, consulte [Sobre listas de referência de patches](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [GetPatchBaseline](#) na Referência de AWS CLI Comandos.

get-service-setting

O código de exemplo a seguir mostra como usar `get-service-setting`.

AWS CLI

Para recuperar a configuração do serviço para a taxa de transferência do Parameter Store

A seguir `get-service-setting` Este exemplo recupera a configuração de serviço atual para a taxa de transferência do Parameter Store na região especificada.

```
aws ssm get-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Aumentando a taxa de transferência do Parameter Store](#) no AWS Systems Manager Users Guide.

- Para API obter detalhes, consulte [.GetServiceSetting](#) na Referência de AWS CLI Comandos.

label-parameter-version

O código de exemplo a seguir mostra como usar `label-parameter-version`.

AWS CLI

Exemplo 1: Para adicionar um rótulo à versão mais recente de um parâmetro

O `label-parameter-version` exemplo a seguir adiciona um rótulo à versão mais recente do parâmetro especificado.

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady"
```

Saída:

```
{  
  "InvalidLabels": [],  
  "ParameterVersion": 3  
}
```

Para obter mais informações, consulte [Trabalhar com rótulos de parâmetros](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: Para adicionar um rótulo a uma versão específica de um parâmetro

O `label-parameter-version` exemplo a seguir adiciona um rótulo à versão especificada de um parâmetro.

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady" \  
  --parameter-version "2" --labels "DevelopmentReady"
```

Para obter mais informações, consulte [Trabalhar com rótulos de parâmetros](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [LabelParameterVersion](#) na Referência de AWS CLI Comandos.

list-association-versions

O código de exemplo a seguir mostra como usar `list-association-versions`.

AWS CLI

Para listar todas as versões de uma associação de um ID de associação específico

O exemplo de `list-association-versions` a seguir lista todas as versões das associações especificadas.

```
aws ssm list-association-versions \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

```
}
```

Para obter mais informações, consulte [Trabalhar com associações no Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ListAssociationVersions](#) na Referência de AWS CLI Comandos.

list-associations

O código de exemplo a seguir mostra como usar `list-associations`.

AWS CLI

Exemplo 1: listar suas associações para uma instância específica

O exemplo de associações de lista a seguir lista todas as associações com `AssociationName`, `UpdateSSMAgent`

```
aws ssm list-associations /  
--association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

Saída:

```
{  
  "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"
}
]
}

```

Para obter mais informações, consulte [Trabalhar com associações no Systems Manager](#) no Guia do usuário do Systems Manager.

Exemplo 2: listar suas associações para um documento específico

O exemplo de associações de lista a seguir lista todas as associações do documento especificado.

```

aws ssm list-associations /
--association-filter-list "key=Name,value=AWS-UpdateSSMAgent"

```

Saída:

```

{
  "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
            }
        }
    }
]
}

```

Para obter mais informações, consulte [Trabalhar com associações no Systems Manager](#) no Guia do usuário do Systems Manager.

- Para API obter detalhes, consulte [ListAssociations](#) na Referência de AWS CLI Comandos.

list-command-invocations

O código de exemplo a seguir mostra como usar `list-command-invocations`.

AWS CLI

Para listar as invocações de um comando específico

O exemplo de `list-command-invocations` a seguir lista todas as invocações de um comando.

```
aws ssm list-command-invocations \
  --command-id "ef7fd8-9b57-4151-a15c-db9a12345678" \
  --details
```

Saída:

```
{
  "CommandInvocations": [
    {
      "CommandId": "ef7fd8-9b57-4151-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
  }
}
]
}

```

Para obter mais informações, consulte [Entender os status dos comandos](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ListCommandInvocations](#) na Referência de AWS CLI Comandos.

list-commands

O código de exemplo a seguir mostra como usar `list-commands`.

AWS CLI

Exemplo 1: obter o status de um comando específico

O exemplo de `list-commands` a seguir recupera e exibe o status do comando especificado.

```

aws ssm list-commands \
  --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"

```

Exemplo 2: para obter o status dos comandos solicitados após uma data específica

O exemplo de `list-commands` a seguir recupera os detalhes dos comandos solicitados após a data especificada.

```
aws ssm list-commands \  
  --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Exemplo 3: Para listar todos os comandos solicitados em uma AWS conta

O `list-commands` exemplo a seguir lista todos os comandos solicitados pelos usuários na AWS conta atual e na região.

```
aws ssm list-commands
```

Saída:

```
{  
  "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
}
}
]
}

```

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ListCommands](#) na Referência de AWS CLI Comandos.

list-compliance-items

O código de exemplo a seguir mostra como usar `list-compliance-items`.

AWS CLI

Para listar itens de conformidade para uma instância específica

Este exemplo lista todos os itens de conformidade para a instância especificada.

Comando:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-  
types "ManagedInstance"
```

Saída:

```
{  
  "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--"
}

```

Para listar itens de conformidade para uma instância e um ID de associação específicos

Este exemplo lista todos os itens de conformidade para a instância e o ID de associação especificados.

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"

```

Para listar itens de conformidade para uma instância específica após uma data e uma hora específicas

Este exemplo lista todos os itens de conformidade para uma instância após a data e a hora especificadas.

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"

```

- Para API obter detalhes, consulte [ListComplianceItems](#) na Referência de AWS CLI Comandos.

list-compliance-summaries

O código de exemplo a seguir mostra como usar `list-compliance-summaries`.

AWS CLI

Para listar resumos de conformidade para todos os tipos de conformidade

Este exemplo lista resumos de conformidade para todos os tipos de conformidade em sua conta.

Comando:

```
aws ssm list-compliance-summaries
```

Saída:

```
{
  "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": 1
    }
},
"NonCompliantSummary": {
    "NonCompliantCount": 1,
    "SeveritySummary": {
        "CriticalCount": 1,
        "HighCount": 0,
        "MediumCount": 0,
        "LowCount": 0,
        "InformationalCount": 0,
        "UnspecifiedCount": 0
    }
}
},
...
],
"NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

Para listar resumos de conformidade para um tipo de conformidade específico

Este exemplo lista o resumo de conformidade para o tipo de conformidade do patch.

Comando:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Para API obter detalhes, consulte [ListComplianceSummaries](#) na Referência de AWS CLI Comandos.

list-document-metadata-history

O código de exemplo a seguir mostra como usar `list-document-metadata-history`.

AWS CLI

Exemplo: Para visualizar o histórico e o status de aprovação de um modelo de alteração

O `list-document-metadata-history` exemplo a seguir retorna o histórico de aprovação do modelo de alteração do Change Manager especificado.

```
aws ssm list-document-metadata-history \
  --name MyChangeManageTemplate \
  --metadata DocumentReviews
```

Saída:

```
{
  "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"
      }
    ]
  }
}
```

Para obter mais informações, consulte [Revisando e aprovando ou rejeitando modelos de alteração](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ListDocumentMetadataHistory](#) na Referência de AWS CLI Comandos.

list-document-versions

O código de exemplo a seguir mostra como usar `list-document-versions`.

AWS CLI

Para listar versões de documentos

O exemplo de `list-document-versions` a seguir lista todas as versões de um documento do Systems Manager.

```
aws ssm list-document-versions \  
  --name "Example"
```

Saída:

```
{  
  "DocumentVersions": [  
    {  
      "Name": "Example",  
      "DocumentVersion": "1",  
      "CreateDate": 1583257938.266,  
      "IsDefaultVersion": true,  
      "DocumentFormat": "YAML",  
      "Status": "Active"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Enviar comandos que usam o parâmetro de versão do documento](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ListDocumentVersions](#) na Referência de AWS CLI Comandos.

list-documents

O código de exemplo a seguir mostra como usar `list-documents`.

AWS CLI

Exemplo 1: listar documentos

O exemplo de `list-documents` a seguir lista documentos pertencentes à conta solicitante marcados com a tag personalizada.

```
aws ssm list-documents \  
  --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Saída:

```
{  
  "DocumentIdentifiers": [  
    {  
      "Name": "Example",  
      "Owner": "29884EXAMPLE",  
      "PlatformTypes": [  
        "Windows",  
        "Linux"  
      ],  
      "DocumentVersion": "1",  
      "DocumentType": "Automation",  
      "SchemaVersion": "0.3",  
      "DocumentFormat": "YAML",  
      "Tags": [  
        {  
          "Key": "DocUse",  
          "Value": "Testing"  
        }  
      ]  
    }  
  ]  
}
```

Para obter mais informações, consulte [Documentos do AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: listar documentos compartilhados

O `list-documents` exemplo a seguir lista documentos compartilhados, incluindo documentos compartilhados privados que não são de propriedade da AWS.

```
aws ssm list-documents \  
  --filters Key=Name,Values=sharedDocNamePrefix Key=Owner,Values=Private
```

Saída:

```
{
  "DocumentIdentifiers": [
    {
      "Name": "Example",
      "Owner": "12345EXAMPLE",
      "PlatformTypes": [
        "Windows",
        "Linux"
      ],
      "DocumentVersion": "1",
      "DocumentType": "Command",
      "SchemaVersion": "0.3",
      "DocumentFormat": "YAML",
      "Tags": []
    }
  ]
}
```

Para obter mais informações, consulte [Documentos do AWS Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ListDocuments](#) na Referência de AWS CLI Comandos.

list-inventory-entries

O código de exemplo a seguir mostra como usar `list-inventory-entries`.

AWS CLI

Exemplo 1: visualizar entradas específicas do tipo de inventário para uma instância

O `list-inventory-entries` exemplo a seguir lista as entradas de inventário do tipo de inventário:Application AWS em uma instância específica.

```
aws ssm list-inventory-entries \
  --instance-id "i-1234567890abcdef0" \
  --type-name "AWS:Application"
```

Saída:

```
{
  "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"
    }
  ]
}
```

Exemplo 2: visualizar entradas de inventário personalizadas atribuídas a uma instância

O exemplo de `list-inventory-entries` a seguir lista uma entrada de inventário personalizada atribuída a uma instância.

```
aws ssm list-inventory-entries \
  --instance-id "i-1234567890abcdef0" \
  --type-name "Custom:RackInfo"
```

Saída:

```
{
  "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"
    }
  ]
}
```



```

    }
  ]
}

```

- Para API obter detalhes, consulte [ListInventoryEntries](#) na Referência de AWS CLI Comandos.

list-ops-item-related-items

O código de exemplo a seguir mostra como usar `list-ops-item-related-items`.

AWS CLI

Para listar os recursos de itens relacionados de um OpsItem

O `list-ops-item-related-items` exemplo a seguir lista os recursos de itens relacionados de um. OpsItem

```

aws ssm list-ops-item-related-items \
  --ops-item-id "oi-f99f2EXAMPLE"

```

Saída:

```

{
  "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"
    }
  ]
}

```

```
]
}
```

Para obter mais informações, consulte [Trabalhando com incidentes do Incident Manager OpsCenter no Guia do Usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [ListOpsItemRelatedItems](#) na Referência de AWS CLI Comandos.

list-resource-compliance-summaries

O código de exemplo a seguir mostra como usar `list-resource-compliance-summaries`.

AWS CLI

Para listar as contagens resumidas de conformidade em nível de recurso

Este exemplo lista as contagens resumidas de conformidade em nível de recurso.

Comando:

```
aws ssm list-resource-compliance-summaries
```

Saída:

```
{
  "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--"
}

```

Para listar resumos de conformidade em nível de recurso para um tipo de conformidade específico

Este exemplo lista resumos de conformidade em nível de recurso para o tipo de conformidade do patch.

Comando:

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Para API obter detalhes, consulte [ListResourceComplianceSummaries](#) na Referência de AWS CLI Comandos.

list-resource-data-sync

O código de exemplo a seguir mostra como usar `list-resource-data-sync`.

AWS CLI

Para listar suas configurações de sincronização de dados de recursos

Este exemplo recupera informações sobre suas configurações de sincronização de dados de recursos.

```
aws ssm list-resource-data-sync
```

Saída:

```
{
  "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"
}
]
```

- Para API obter detalhes, consulte [ListResourceDataSync](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags aplicadas a uma lista de referência de patches

O exemplo de `list-tags-for-resource` a seguir lista as tags para uma lista de referência de patches.

```
aws ssm list-tags-for-resource \
  --resource-type "PatchBaseline" \
  --resource-id "pb-0123456789abcdef0"
```

Saída:

```
{
  "TagList": [
    {
      "Key": "Environment",
      "Value": "Production"
    }
  ],
```

```
{
  "Key": "Region",
  "Value": "EMEA"
}
]
```

Para obter mais informações, consulte [AWS Recursos de marcação](#) na Referência AWS geral.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

modify-document-permission

O código de exemplo a seguir mostra como usar `modify-document-permission`.

AWS CLI

Para modificar as permissões do documento

O exemplo de `modify-document-permission` a seguir compartilha publicamente um documento do Systems Manager.

```
aws ssm modify-document-permission \
  --name "Example" \
  --permission-type "Share" \
  --account-ids-to-add "ALL"
```

Este comando não produz saída.

Para obter mais informações, consulte [Compartilhar um documento do Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ModifyDocumentPermission](#) na Referência de AWS CLI Comandos.

put-compliance-items

O código de exemplo a seguir mostra como usar `put-compliance-items`.

AWS CLI

Para registrar um tipo de conformidade e detalhes de conformidade em uma instância designada

Este exemplo registra o tipo de conformidade Custom:AVCheck na instância gerenciada especificada. Não haverá saída se o comando for bem-sucedido.

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"
```

- Para API obter detalhes, consulte [PutComplianceItems](#) na Referência de AWS CLI Comandos.

put-inventory

O código de exemplo a seguir mostra como usar put-inventory.

AWS CLI

Para atribuir metadados de inventário personalizados a uma instância

Este exemplo atribui informações de localização de rack a uma instância. Não haverá saída se o comando for bem-sucedido.

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']"
```

- Para API obter detalhes, consulte [PutInventory](#) na Referência de AWS CLI Comandos.

put-parameter

O código de exemplo a seguir mostra como usar put-parameter.

AWS CLI

Exemplo 1: como alterar o valor de um parâmetro

O exemplo de `put-parameter` a seguir altera o valor do parâmetro especificado.

```
aws ssm put-parameter \  
  --name "MyStringParameter" \  
  --type "String" \  
  --value "Vici" \  
  --overwrite
```

Saída:

```
{  
  "Version": 2,  
  "Tier": "Standard"  
}
```

[Para obter mais informações, consulte `Create a Systems Manager parameter \(AWS CLI\)`, `'Gerenciando camadas de parâmetros < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ e Trabalhando com políticas de parâmetros no Systems Manager User Guide.AWS`](#)

Exemplo 2: como criar um parâmetro avançado

O exemplo de `put-parameter` a seguir cria um parâmetro avançado.

```
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
```

Saída:

```
{
```



```
"Version": 1,  
"Tier": "Advanced"  
}
```

[Para obter mais informações, consulte Create a Systems Manager parameter \(AWS CLI\), 'Gerenciando camadas de parâmetros < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ e Trabalhando com políticas de parâmetros no Systems Manager User Guide.AWS](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html)

Exemplo 3: como converter um parâmetro padrão em um parâmetro avançado

O exemplo de `put-parameter` a seguir converte um parâmetro padrão existente em um parâmetro avançado.

```
aws ssm put-parameter \  
  --name "MyConvertedParameter" \  
  --value "abc123" \  
  --type "String" \  
  --tier Advanced \  
  --overwrite
```

Saída:

```
{  
  "Version": 2,  
  "Tier": "Advanced"  
}
```

[Para obter mais informações, consulte Create a Systems Manager parameter \(AWS CLI\), 'Gerenciando camadas de parâmetros < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ e Trabalhando com políticas de parâmetros no Systems Manager User Guide.AWS](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html)

Exemplo 4: como criar um parâmetro com uma política anexada

O exemplo de `put-parameter` a seguir cria um parâmetro avançado com uma política de parâmetros anexada.

```
aws ssm put-parameter \  
  --name "/Finance/Payroll/q2accesskey" \  
  --policy "arn:aws:iam::123456789012:policy/FinancePayrollAccessKeyPolicy"
```

```

--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"}}]"

```

Saída:

```

{
  "Version": 1,
  "Tier": "Advanced"
}

```

[Para obter mais informações, consulte Create a Systems Manager parameter \(AWS CLI\), 'Gerenciando camadas de parâmetros < https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html> e Trabalhando com políticas de parâmetros no Systems Manager User Guide.AWS](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html)

Exemplo 5: como adicionar uma política a um parâmetro existente

O exemplo de `put-parameter` a seguir anexa uma política a um parâmetro avançado existente.

```

aws ssm put-parameter \
--name "/Finance/Payroll/q2accesskey" \
--value "N3wP@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"}}]"
--overwrite

```

Saída:

```

{
  "Version": 2,

```

```
"Tier": "Advanced"
}
```

Para obter mais informações, consulte [Create a Systems Manager parameter \(AWS CLI\)](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html), [Gerenciando camadas de parâmetros < https://docs.aws.amazon.com/systems-manager/ latest/ userguide/ parameter-store-advanced-parameters .html>](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html) e [Trabalhando com políticas de parâmetros no Systems Manager User Guide.AWS](#)

- Para API obter detalhes, consulte [PutParameter](#) na Referência de AWS CLI Comandos.

register-default-patch-baseline

O código de exemplo a seguir mostra como usar `register-default-patch-baseline`.

AWS CLI

Para definir a lista de referência de patches padrão

O exemplo de `register-default-patch-baseline` a seguir registra a lista de referência de patches personalizada especificada como a lista de referência de patches padrão para o tipo de sistema operacional ao qual ela oferece suporte.

```
aws ssm register-default-patch-baseline \
  --baseline-id "pb-abc123cf9bEXAMPLE"
```

Saída:

```
{
  "BaselineId": "pb-abc123cf9bEXAMPLE"
}
```

O `register-default-patch-baseline` exemplo a seguir registra a linha de base do patch padrão fornecida pelo AWS CentOS como a linha de base do patch padrão.

```
aws ssm register-default-patch-baseline \
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
  pb-0574b43a65ea646ed"
```

Saída:

```
{
  "BaselineId": "pb-abc123cf9bEXAMPLE"
}
```

Para obter mais informações, consulte [Sobre listas de referência de patches predefinidas e personalizadas](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [RegisterDefaultPatchBaseline](#) na Referência de AWS CLI Comandos.

register-patch-baseline-for-patch-group

O código de exemplo a seguir mostra como usar `register-patch-baseline-for-patch-group`.

AWS CLI

Para registrar uma lista de referência de patches de um grupo de patches

O exemplo de `register-patch-baseline-for-patch-group` a seguir registra uma lista de referência de patches para um grupo de patches.

```
aws ssm register-patch-baseline-for-patch-group \
  --baseline-id "pb-045f10b4f382baeda" \
  --patch-group "Production"
```

Saída:

```
{
  "BaselineId": "pb-045f10b4f382baeda",
  "PatchGroup": "Production"
}
```

Para obter mais informações, consulte Criar um grupo de patches < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html> > e [Adicionar um grupo de patches a uma linha de base de patches no Guia do usuário do Systems Manager.AWS](#)

- Para API obter detalhes, consulte [RegisterPatchBaselineForPatchGroup](#) na Referência de AWS CLI Comandos.

register-target-with-maintenance-window

O código de exemplo a seguir mostra como usar `register-target-with-maintenance-window`.

AWS CLI

Exemplo 1: registrar um único destino com uma janela de manutenção

O exemplo de `register-target-with-maintenance-window` a seguir registra uma instância com uma janela de manutenção.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \  
  --owner-information "Single instance" \  
  --resource-type "INSTANCE"
```

Saída:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Exemplo 2: Para registrar vários destinos com uma janela de manutenção usando a instância IDs

O `register-target-with-maintenance-window` exemplo a seguir registra duas instâncias com uma janela de manutenção especificando sua instância. 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"
```

Saída:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Exemplo 3: registrar destinos com uma janela de manutenção usando tags de recursos

O exemplo de `register-target-with-maintenance-window` a seguir registra instâncias com uma janela de manutenção especificando tags de recursos que foram aplicadas às instâncias.

```
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"
```

Saída:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Exemplo 4: registrar destinos usando um grupo de chaves de tag

O exemplo de `register-target-with-maintenance-window` a seguir registra instâncias que têm uma ou mais chaves de tags atribuídas a elas, independentemente de seus valores de chave.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "INSTANCE" \  
  --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

Saída:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Exemplo 5: registrar destinos usando um nome de grupo de recursos

O exemplo de `register-target-with-maintenance-window` a seguir registra um grupo de recursos especificado, independentemente do tipo de recurso que ele contém.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "RESOURCE_GROUP" \  
  --target "Name=ExampleResourceGroup"
```

```
--target "Key=resource-groups:Name,Values=MyResourceGroup"
```

Saída:

```
{
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Para obter mais informações, consulte [Registrar uma instância de destino com a janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [RegisterTargetWithMaintenanceWindow](#) na Referência de AWS CLI Comandos.

register-task-with-maintenance-window

O código de exemplo a seguir mostra como usar `register-task-with-maintenance-window`.

AWS CLI

Exemplo 1: registrar uma tarefa do Automation com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Automation com uma janela de manutenção voltada para uma instância.

```
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"
```

Saída:

```
{
  "WindowTaskId": "11144444-5555-6666-7777-88888888"
}
```

```
}

```

Para obter mais informações, consulte [Registrar uma tarefa com a janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: registrar uma tarefa do Lambda com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Lambda com uma janela de manutenção voltada para uma instância.

```
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"
```

Saída:

```
{
  "WindowTaskId": "22244444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte [Registrar uma tarefa com a janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

Exemplo 3: registrar uma tarefa do Run Command com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Run Command com uma janela de manutenção voltada para uma instância.

```
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/EZ0ut-33ae0fb7/file/110351/1/EZ0ut.zip\\\"],
\\\"workingDirectory\\\":[\\\"\\\\\\\\\\\"]},\\\"TimeoutSeconds\\\":600}}\" \
--max-concurrency 1 \
--max-errors 1 \
--priority 10

```

Saída:

```

{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}

```

Para obter mais informações, consulte [Registrar uma tarefa com a janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

Exemplo 4: registrar uma tarefa do Step Functions com uma janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa do Step Functions com uma janela de manutenção voltada para uma instância.

```

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"

```

Saída:

```
{
  "WindowTaskId": "44444444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte [Registrar uma tarefa com a janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

Exemplo 5: registrar uma tarefa usando um ID de destino de janela de manutenção

O exemplo de `register-task-with-maintenance-window` a seguir registra uma tarefa usando um ID de destino de janela de manutenção. O ID de destino da janela de manutenção estava presente na saída do comando `aws ssm register-target-with-maintenance-window`. Também é possível recuperá-lo da saída do comando `aws ssm describe-maintenance-window-targets`.

```
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
```

Saída:

```
{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}
```

Para obter mais informações, consulte [Registrar uma tarefa com a janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [RegisterTaskWithMaintenanceWindow](#) na Referência de AWS CLI Comandos.

remove-tags-from-resource

O código de exemplo a seguir mostra como usar `remove-tags-from-resource`.

AWS CLI

Remover uma tag de uma lista de referência de patches

O exemplo de `remove-tags-from-resource` a seguir remove duas tags de uma lista de referência de patches.

```
aws ssm remove-tags-from-resource \  
  --resource-type "PatchBaseline" \  
  --resource-id "pb-0123456789abcdef0" \  
  --tag-keys "Region"
```

Este comando não produz saída.

Para obter mais informações, consulte [AWS Recursos de marcação](#) na Referência AWS geral.

- Para API obter detalhes, consulte [RemoveTagsFromResource](#) na Referência de AWS CLI Comandos.

reset-service-setting

O código de exemplo a seguir mostra como usar `reset-service-setting`.

AWS CLI

Para redefinir a configuração do serviço para a taxa de transferência do Parameter Store

O `reset-service-setting` exemplo a seguir redefine a configuração do serviço para a taxa de transferência do Parameter Store na região especificada para não usar mais a taxa de transferência aumentada.

```
aws ssm reset-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Aumentando a taxa de transferência do Parameter Store](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [ResetServiceSetting](#) na Referência de AWS CLI Comandos.

resume-session

O código de exemplo a seguir mostra como usar `resume-session`.

AWS CLI

Para retomar uma sessão do Gerenciador de Sessões

Este `resume-session` exemplo retoma uma sessão do Gerenciador de Sessões com uma instância depois que ela é desconectada. Observe que esse comando interativo exige que o plug-in do Session Manager seja instalado na máquina-cliente que faz a chamada.

```

aws ssm resume-session \
  --session-id Mary-Major-07a16060613c408b5

```

Saída:

```

{
  "SessionId": "Mary-Major-07a16060613c408b5",
  "TokenValue":
  "AAEAAVbTGsa0nyvcUoNGqifbv5r/81gxuQ1jCuY8qVcv0noBAAAAAFxtd3jIXAFUUXGTJ7zF/
AWJpWdvi0lF5p3dlAgrqVIV06IEXhkHLz0/1gXKRKEME71E6TL0p1LDJAMZ
+kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wju1PLex08SH17g5R/
wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11D0IT2doz0qXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd
+ihfGa6MEJJ97Jmat/a2TspEn0jNn9Mvu5iwXIW2yCvWZrGUj+/
QI5Xr7s1XJBEEnSKR54o4fN0GV9RWl0RZsZm1mki0JJtiwWgZ",
  "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-
Major-07a16060613c408b5?role=publish_subscribe"
}

```

Para obter mais informações, consulte [Instalar o plug-in do Gerenciador de Sessões AWS CLI no Guia do Usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [ResumeSession](#) na Referência de AWS CLI Comandos.

send-automation-signal

O código de exemplo a seguir mostra como usar `send-automation-signal`.

AWS CLI

Para enviar um sinal para uma execução de automação

O `send-automation-signal` exemplo a seguir envia um sinal de aprovação para uma execução de automação.

```
aws ssm send-automation-signal \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \  
  --signal-type "Approve"
```

Este comando não produz saída.

Para obter mais informações, consulte [Executando um fluxo de trabalho de automação com aprovadores](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [SendAutomationSignal](#) na Referência de AWS CLI Comandos.

send-command

O código de exemplo a seguir mostra como usar `send-command`.

AWS CLI

Exemplo 1: executar um comando em uma ou mais instâncias remotas

O exemplo de `send-command` a seguir executa um comando `echo` em uma instância de destino.

```
aws ssm send-command \  
  --document-name "AWS-RunShellScript" \  
  --parameters 'commands=["echo HelloWorld"]' \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0" \  
  --
```

```
--comment "echo HelloWorld"
```

Saída:

```
{
  "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
    }
  }
}
```

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: obter informações de IP sobre uma instância

O exemplo de send-command a seguir retorna as informações de IP sobre uma instância.

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig"
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 3: executar um comando em instâncias com tags específicas

O send-command exemplo a seguir executa um comando em instâncias que têm a chave de tag ENV "" e o valor "Dev".

```
aws ssm send-command \  
  --targets "Key=tag:ENV,Values=Dev" \  
  --document-name "AWS-RunShellScript" \  
  --parameters "commands=ifconfig"
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 4: Para executar um comando que envia SNS notificações

O send-command exemplo a seguir executa um comando que envia SNS notificações para todos os eventos de notificação e o tipo de Command notificação.

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig" \  
  --notification-configuration "snsTopic=arn:aws:sns:us-east-1:123456789012:my-topic"
```

```
--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"
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 5: Para executar um comando que retorna para S3 e CloudWatch

O send-command exemplo a seguir executa um comando que envia detalhes do comando para um bucket do S3 e para um grupo de registros de CloudWatch registros.

```
aws ssm send-command \  
--instance-ids "i-1234567890abcdef0" \  
--document-name "AWS-RunShellScript" \  
--comment "IP config" \  
--parameters "commands=ifconfig" \  
--output-s3-bucket-name "s3-bucket-name" \  
--output-s3-key-prefix "runcommand" \  
--cloud-watch-output-  
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 6: executar comandos em várias instâncias com tags diferentes

O exemplo de send-command a seguir executa um comando em instâncias com duas chaves e valores de tag diferentes.

```
aws ssm send-command \  
--document-name "AWS-RunPowerShellScript" \  
--parameters commands=["echo helloWorld"] \  
--targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 7: usar várias instâncias com a mesma chave de tag

O exemplo de `send-command` a seguir executa um comando em instâncias que têm a mesma chave de tag, mas com valores diferentes.

```
aws ssm send-command \  
  --document-name "AWS-RunPowerShellScript" \  
  --parameters commands=["echo helloWorld"] \  
  --targets Key=tag:Env,Values=Dev,Test
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Executar comandos usando o Systems Manager Run Command](#) no Guia do usuário do AWS Systems Manager.

Exemplo 8: executar um comando que usa um documento compartilhado

O exemplo de `send-command` a seguir executa um comando compartilhado em uma instância de destino.

```
aws ssm send-command \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Consulte um exemplo de saída no exemplo 1.

Para obter mais informações, consulte [Usando SSM documentos compartilhados](#) no AWS Systems Manager User Guide.

- Para API obter detalhes, consulte [SendCommand](#) na Referência de AWS CLI Comandos.

start-associations-once

O código de exemplo a seguir mostra como usar `start-associations-once`.

AWS CLI

Para administrar uma associação imediatamente e somente uma vez

O `start-associations-once` exemplo a seguir executa a associação especificada imediatamente e somente uma vez. Não haverá saída se o comando for bem-sucedido.

```
aws ssm start-associations-once \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Este comando não produz saída.

Para obter mais informações, consulte [Visualizar históricos de associações](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [StartAssociationsOnce](#) na Referência de AWS CLI Comandos.

start-automation-execution

O código de exemplo a seguir mostra como usar `start-automation-execution`.

AWS CLI

Exemplo 1: executar um documento do Automation

O exemplo de `start-automation-execution` a seguir executa um documento do Automation.

```
aws ssm start-automation-execution \  
  --document-name "AWS-UpdateLinuxAmi" \  
  --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/  
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Saída:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

Para obter mais informações, consulte [Executar um fluxo de trabalho do Automation manualmente](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: executar um documento do Automation compartilhado

O exemplo de `start-automation-execution` a seguir executa um documento do Automation compartilhado.

```
aws ssm start-automation-execution \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Saída:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

Para obter mais informações, consulte [Usando SSM documentos compartilhados](#) no AWS Systems Manager User Guide.

- Para API obter detalhes, consulte [StartAutomationExecution](#) na Referência de AWS CLI Comandos.

start-change-request-execution

O código de exemplo a seguir mostra como usar start-change-request-execution.

AWS CLI

Exemplo 1: Para iniciar uma solicitação de alteração

O start-change-request-execution exemplo a seguir inicia uma solicitação de alteração com o mínimo de opções especificadas.

```
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"
```

Saída:

```
{  
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"  
}
```

Exemplo 2: Para iniciar uma solicitação de alteração usando um JSON arquivo externo

O `start-automation-execution` exemplo a seguir inicia uma solicitação de alteração com várias opções especificadas em um JSON arquivo.

```
aws ssm start-change-request-execution \  
--cli-input-json file://MyChangeRequest.json
```

Conteúdo de `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?\nThis change template demonstrates the feature set available
for creating change templates for Change Manager. This template starts a Runbook
workflow for the Automation document called AWS-HelloWorld.\n\n## Input Parameters
\n* ApproverSnsTopicArn: (Required) Amazon Simple Notification Service ARN for
approvers.\n* Approver: (Required) The name of the approver to send this request
to.\n* ApproverType: (Required) The type of reviewer.\n * Allowed Values: IamUser,
IamGroup, IamRole, SSOGroup, SSOUser\n\n## Output Parameters\nThis document has no
outputs \n"
}

```

Saída:

```

{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}

```

Para obter mais informações, consulte [Criação de solicitações de alteração](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [StartChangeRequestExecution](#) na Referência de AWS CLI Comandos.

start-session

O código de exemplo a seguir mostra como usar `start-session`.

AWS CLI

Exemplo 1: Para iniciar uma sessão do Session Manager

Este exemplo de `start-session` estabelece uma conexão com uma instância para uma sessão do Session Manager. Observe que esse comando interativo exige que o plug-in do Session Manager seja instalado na máquina-cliente que faz a chamada.

```

aws ssm start-session \
  --target "i-1234567890abcdef0"

```

Saída:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

Exemplo 2: Para iniciar uma sessão do Gerenciador de Sessões usando SSH

Este `start-session` exemplo estabelece uma conexão com uma instância para uma sessão do Gerenciador de Sessões usando SSH. Observe que esse comando interativo exige que o plug-in do Gerenciador de Sessões seja instalado na máquina cliente que faz a chamada e que o comando use o usuário padrão na instância, como `ec2-user` para EC2 instâncias do Linux.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

Saída:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

Para obter mais informações, consulte [Iniciar uma sessão](#) e [instalar o plug-in do gerenciador de sessões AWS CLI no Guia do usuário do AWS Systems Manager](#).

- Para API obter detalhes, consulte [StartSession](#) na Referência de AWS CLI Comandos.

stop-automation-execution

O código de exemplo a seguir mostra como usar `stop-automation-execution`.

AWS CLI

Para interromper a execução de uma automação

O exemplo de `stop-automation-execution` a seguir interrompe um documento do Automation.

```
aws ssm stop-automation-execution  
--automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Este comando não produz saída.

Para obter mais informações, consulte [Executar um fluxo de trabalho do Automation manualmente](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [StopAutomationExecution](#) na Referência de AWS CLI Comandos.

terminate-session

O código de exemplo a seguir mostra como usar `terminate-session`.

AWS CLI

Para encerrar uma sessão do Gerenciador de Sessões

Esse `terminate-session` exemplo encerra permanentemente uma sessão criada pelo usuário “Shirley-Rodriguez” e fecha a conexão de dados entre o cliente do Gerenciador de Sessões e o Agente na instância. SSM

```
aws ssm terminate-session \  
  --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

Saída:

```
{  
  "SessionId": "Shirley-Rodriguez-07a16060613c408b5"  
}
```

Para obter mais informações, consulte [Encerrar uma sessão](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [TerminateSession](#) na Referência de AWS CLI Comandos.

unlabel-parameter-version

O código de exemplo a seguir mostra como usar `unlabel-parameter-version`.

AWS CLI

Para excluir rótulos de parâmetros

O `unlabel-parameter-version` exemplo a seguir exclui os rótulos especificados da versão do parâmetro fornecido.

```
aws ssm unlabel-parameter-version \  
  --name "parameterName" \  
  --parameter-version "version" \  
  --labels "label_1" "label_2" "label_3"
```

Saída:

```
{
  "RemovedLabels": [
    "label_1"
    "label_2"
    "label_3"
  ],
  "InvalidLabels": []
}
```

Para obter mais informações, consulte [Excluir rótulos de parâmetros \(AWS CLI\)](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UnlabelParameterVersion](#) na Referência de AWS CLI Comandos.

update-association-status

O código de exemplo a seguir mostra como usar update-association-status.

AWS CLI

Para atualizar o status da associação

O exemplo de update-association-status a seguir atualiza o status da associação entre uma instância e um 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=AdditionalInfo,Config-Needed"
```

Saída:

```
{
  "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
  }
}

```

Para obter mais informações, consulte [Trabalhar com associações no Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateAssociationStatus](#) na Referência de AWS CLI Comandos.

update-association

O código de exemplo a seguir mostra como usar update-association.

AWS CLI

Exemplo 1: atualizar uma associação de documentos

O exemplo de `update-association` a seguir atualiza uma associação com uma nova versão de documento.

```
aws ssm update-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \  
  --document-version "$LATEST"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Editar e criar uma nova versão de uma associação](#) no Guia do usuário do AWS Systems Manager.

Exemplo 2: atualizar a expressão de programação de uma associação

O exemplo de `update-association` a seguir atualiza a expressão de programação para a associação especificada.

```
aws ssm update-association \  
--association-id "8dfe3659-4309-493a-8755-0123456789ab" \  
--schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Editar e criar uma nova versão de uma associação](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateAssociation](#) na Referência de AWS CLI Comandos.

update-document-default-version

O código de exemplo a seguir mostra como usar `update-document-default-version`.

AWS CLI

Para atualizar a versão padrão de um documento

O exemplo de `update-document-default-version` a seguir atualiza a versão padrão de um documento do Systems Manager.

```
aws ssm update-document-default-version \  
  --name "Example" \  
  --document-version "2"
```

Saída:

```
{  
  "Description": {  
    "Name": "Example",  
    "DefaultVersion": "2"  
  }  
}
```

Para obter mais informações, consulte [Escrevendo conteúdo de SSM documentos](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateDocumentDefaultVersion](#) na Referência de AWS CLI Comandos.

update-document-metadata

O código de exemplo a seguir mostra como usar `update-document-metadata`.

AWS CLI

Exemplo: Para aprovar a versão mais recente de um modelo de alteração

O seguinte `update-document-metadata` fornece uma aprovação para a versão mais recente de um modelo de alteração que foi enviado para análise.

```
aws ssm update-document-metadata \  
  --name MyChangeManagerTemplate \  
  --document-reviews 'Action=Approve, Comment=[{Type=Comment, Content=Approved!}]'
```

Este comando não produz saída.

Para obter mais informações, consulte [Revisando e aprovando ou rejeitando modelos de alteração](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateDocumentMetadata](#) na Referência de AWS CLI Comandos.

update-document

O código de exemplo a seguir mostra como usar update-document.

AWS CLI

Para criar uma nova versão de um documento

O exemplo de update-document a seguir cria uma nova versão de um documento quando executado em um computador Windows. O documento especificado por --document deve estar no JSON formato. Observe que file:// deve ser referenciado seguido pelo caminho do arquivo de conteúdo. Em função disso do \$ no início do parâmetro --document-version, o valor deve ser colocado entre aspas duplas no Windows. No Linux, macOS ou em um PowerShell prompt, você deve colocar o valor entre aspas simples.

Versão do Windows:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version "$LATEST"
```

Versão do Linux/Mac:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version '$LATEST'
```

Saída:

```
{  
  "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"
  }
}

```

- Para API obter detalhes, consulte [UpdateDocument](#) na Referência de AWS CLI Comandos.

update-maintenance-window-target

O código de exemplo a seguir mostra como usar `update-maintenance-window-target`.

AWS CLI

Para atualizar o alvo de uma janela de manutenção

O `update-maintenance-window-target` exemplo a seguir atualiza somente o nome de um destino da janela de manutenção.

```

aws ssm update-maintenance-window-target \
  --window-id "mw-0c5ed765acEXAMPLE" \
  --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
  --name "NewName" \

```

```
--no-replace
```

Saída:

```
{
  "Description": "",
  "OwnerInformation": "",
  "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
  "WindowId": "mw-0c5ed765acEXAMPLE",
  "Targets": [
    {
      "Values": [
        "i-1234567890EXAMPLE"
      ],
      "Key": "InstanceIds"
    }
  ],
  "Name": "NewName"
}
```

Para obter mais informações, consulte [Atualizar uma janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateMaintenanceWindowTarget](#) na Referência de AWS CLI Comandos.

update-maintenance-window-task

O código de exemplo a seguir mostra como usar update-maintenance-window-task.

AWS CLI

Para atualizar uma tarefa da janela de manutenção

O update-maintenance-window-task exemplo a seguir atualiza a função de serviço para uma tarefa da janela de manutenção.

```
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"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Atualizar uma janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateMaintenanceWindowTask](#) na Referência de AWS CLI Comandos.

update-maintenance-window

O código de exemplo a seguir mostra como usar update-maintenance-window.

AWS CLI

Exemplo 1: atualizar uma janela de manutenção

O exemplo de `update-maintenance-window` a seguir atualiza o nome de uma janela de manutenção.

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --name "My-Renamed-MW"
```

Saída:

```
{  
  "Cutoff": 1,  
  "Name": "My-Renamed-MW",  
  "Schedule": "cron(0 16 ? * TUE *)",  
  "Enabled": true,  
  "AllowUnassociatedTargets": true,  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9",  
  "Duration": 4  
}
```

Exemplo 2: desabilitar uma janela de manutenção

O exemplo de `update-maintenance-window` a seguir desabilita uma janela de manutenção.

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --no-enabled
```

Exemplo 3: habilitar uma janela de manutenção

O exemplo de `update-maintenance-window` a seguir habilita uma janela de manutenção.

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --enabled
```

Para obter mais informações, consulte [Atualizar uma janela de manutenção \(AWS CLI\)](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateMaintenanceWindow](#) na Referência de AWS CLI Comandos.

update-managed-instance-role

O código de exemplo a seguir mostra como usar `update-managed-instance-role`.

AWS CLI

Para atualizar a IAM função de uma instância gerenciada

O `update-managed-instance-role` exemplo a seguir atualiza o IAM perfil de uma instância gerenciada.

```
aws ssm update-managed-instance-role \  
  --instance-id "mi-08ab247cdfEXAMPLE" \  
  --iam-role "ExampleRole"
```

Este comando não produz saída.

Para obter mais informações, consulte [Etapa 4: Criar um perfil de IAM instância para o Systems Manager](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateManagedInstanceRole](#) na Referência de AWS CLI Comandos.

update-ops-item

O código de exemplo a seguir mostra como usar `update-ops-item`.

AWS CLI

Para atualizar um OpsItem

O `update-ops-item` exemplo a seguir atualiza a descrição, a prioridade e a categoria de um OpsItem. Além disso, o comando especifica um SNS tópico para o qual as notificações são enviadas quando isso OpsItem é editado ou alterado.

```
aws ssm update-ops-item \  
  --ops-item-id "oi-287b5EXAMPLE" \  
  --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \  
  --priority 2 \  
  --sns-topic-arn "arn:aws:sns:us-east-1:123456789012:opsitem-failover"
```

```
--category "Security" \  
--notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

Saída:

This command produces no output.

Para obter mais informações, consulte [Trabalhando com OpsItems](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateOpsItem](#) na Referência de AWS CLI Comandos.

update-patch-baseline

O código de exemplo a seguir mostra como usar `update-patch-baseline`.

AWS CLI

Exemplo 1: atualizar uma lista de referência de patches

O exemplo de `update-patch-baseline` a seguir adiciona os dois patches especificados como rejeitados e um patch como aprovado à lista de referência de patches especificada.

```
aws ssm update-patch-baseline \  
  --baseline-id "pb-0123456789abcdef0" \  
  --rejected-patches "KB2032276" "MS10-048" \  
  --approved-patches "KB2124261"
```

Saída:

```
{  
  "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": []
}

```

Exemplo 2: renomear uma lista de referência de patches

O exemplo de `update-patch-baseline` a seguir renomeia lista de referência de patches especificada.

```

aws ssm update-patch-baseline \
  --baseline-id "pb-0713accee01234567" \
  --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"

```

Para obter mais informações, consulte Atualizar ou excluir uma linha de base de patch < <https://docs.aws.amazon.com/systems-manager/latest/userguide/-delete.html>>`__ no Guia patch-baseline-update-or do usuário do Systems Manager.AWS

- Para API obter detalhes, consulte [UpdatePatchBaseline](#) na Referência de AWS CLI Comandos.

update-resource-data-sync

O código de exemplo a seguir mostra como usar `update-resource-data-sync`.

AWS CLI

Para atualizar uma sincronização de dados de recursos

O `update-resource-data-sync` exemplo a seguir atualiza uma sincronização `SyncFromSource` de dados de recursos.

```
aws ssm update-resource-data-sync \  
  --sync-name exampleSync \  
  --sync-type SyncFromSource \  
  --sync-source '{"SourceType":"SingleAccountMultiRegions", "SourceRegions":["us-east-1", "us-west-2"]}'
```

Este comando não produz saída.

Para obter mais informações, consulte [Configurando o Systems Manager Explorer para exibir dados de várias contas e regiões](#) no Guia do usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateResourceDataSync](#) na Referência de AWS CLI Comandos.

update-service-setting

O código de exemplo a seguir mostra como usar `update-service-setting`.

AWS CLI

Para atualizar a configuração do serviço para a taxa de transferência do Parameter Store

O `update-service-setting` exemplo a seguir atualiza a configuração atual do serviço para a taxa de transferência do Parameter Store na região especificada para usar maior taxa de transferência.

```
aws ssm update-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled \  
  --setting-value HighThroughputEnabled
```

```
--setting-value true
```

Este comando não produz saída.

Para obter mais informações, consulte [Aumentando a taxa de transferência do Parameter Store](#) no Guia do Usuário do AWS Systems Manager.

- Para API obter detalhes, consulte [UpdateServiceSetting](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Textract usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Textract.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

analyze-document

O código de exemplo a seguir mostra como usar `analyze-document`.

AWS CLI

Como analisar texto em um documento

O exemplo de `analyze-document` a seguir mostra como analisar texto em um 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
```

Saída:

```
{  
  "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
}
}

```

Para obter mais informações, consulte [Analyzing Document Text with Amazon Textract](#) Guia do desenvolvedor do Amazon Textract

- Para API obter detalhes, consulte [AnalyzeDocument](#) na Referência de AWS CLI Comandos.

detect-document-text

O código de exemplo a seguir mostra como usar detect-document-text.

AWS CLI

Como detectar texto em um documento

O exemplo de detect-document-text a seguir mostra como detectar texto em um documento.

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
```


Saída:

```
{
  "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
}
}
```

Para obter mais informações, consulte [Detecting Document Text with Amazon Textract](#) Guia do desenvolvedor do Amazon Textract

- Para API obter detalhes, consulte [DetectDocumentText](#) na Referência de AWS CLI Comandos.

get-document-analysis

O código de exemplo a seguir mostra como usar `get-document-analysis`.

AWS CLI

Como obter os resultados de uma análise assíncrona de texto em um documento com várias páginas

O exemplo de `get-document-analysis` a seguir mostra como obter os resultados de uma análise assíncrona de texto em um documento com várias páginas.

```
aws textract get-document-analysis \  
  --job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \  
  --max-results 1000
```

Saída:

```
{  
  "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+1VOImoTRmP1zw4P0RFtaeV9Bzhnfedpx1YqwB4xaGDA==",
"DocumentMetadata": {
    "Pages": 1
},
"JobStatus": "SUCCEEDED"
}

```

Para obter mais informações, consulte [Detecting and Analyzing Text in Multi-Page Documents](#) no Guia do desenvolvedor do Amazon Textract

- Para API obter detalhes, consulte [GetDocumentAnalysis](#) na Referência de AWS CLI Comandos.

get-document-text-detection

O código de exemplo a seguir mostra como usar `get-document-text-detection`.


```

        {
            "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"
}

```

Para obter mais informações, consulte [Detecting and Analyzing Text in Multi-Page Documents](#) no Guia do desenvolvedor do Amazon Textract

- Para API obter detalhes, consulte [GetDocumentTextDetection](#) na Referência de AWS CLI Comandos.

start-document-analysis

O código de exemplo a seguir mostra como usar `start-document-analysis`.

AWS CLI

Como começar a analisar texto em um documento com várias páginas

O exemplo de `start-document-analysis` a seguir mostra como iniciar a análise assíncrona de texto em um documento com várias páginas.

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"
```

Saída:

```
{
  "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"
}
```

Para obter mais informações, consulte [Detecting and Analyzing Text in Multi-Page Documents](#) no Guia do desenvolvedor do Amazon Textract

- Para API obter detalhes, consulte [StartDocumentAnalysis](#) na Referência de AWS CLI Comandos.

start-document-text-detection

O código de exemplo a seguir mostra como usar `start-document-text-detection`.

AWS CLI

Como começar a detectar texto em um documento com várias páginas

O exemplo de `start-document-text-detection` a seguir mostra como iniciar a detecção assíncrona de texto em um documento com várias páginas.

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"
```

Saída:

```
{
  "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"
}
```

Para obter mais informações, consulte [Detecting and Analyzing Text in Multi-Page Documents](#) no Guia do desenvolvedor do Amazon Textract

- Para API obter detalhes, consulte [StartDocumentTextDetection](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Transcribe usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Transcribe.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-language-model

O código de exemplo a seguir mostra como usar `create-language-model`.

AWS CLI

Exemplo 1: Para criar um modelo de linguagem personalizado usando dados de treinamento e ajuste.

O `create-language-model` exemplo a seguir cria um modelo de linguagem personalizado. Você pode usar um modelo de linguagem personalizado para melhorar o desempenho da transcrição em domínios como jurídico, hotelaria, finanças e seguros. Para código de idioma, insira um código de idioma válido. Para `base-model-name`, especifique um modelo básico que seja mais adequado para a taxa de amostragem do áudio que você deseja transcrever com seu modelo de idioma personalizado. Para `model-name`, especifique o nome que você deseja chamar de modelo de linguagem personalizado.

```
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"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Improving Domain-Specific Transcription Accuracy with Custom Language Models](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 2: Para criar um modelo de linguagem personalizado usando somente dados de treinamento.

O exemplo de `create-language-model` a seguir transcreve o arquivo de áudio. Você pode usar um modelo de linguagem personalizado para melhorar o desempenho da transcrição em domínios como jurídico, hotelaria, finanças e seguros. Para código de idioma, insira um código de idioma válido. Para `base-model-name`, especifique um modelo básico que seja mais adequado para a taxa de amostragem do áudio que você deseja transcrever com seu modelo de idioma personalizado. Para `model-name`, especifique o nome que você deseja chamar de modelo de linguagem personalizado.

```
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"
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Improving Domain-Specific Transcription Accuracy with Custom Language Models](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [CreateLanguageModel](#) na Referência de AWS CLI Comandos.

create-medical-vocabulary

O código de exemplo a seguir mostra como usar `create-medical-vocabulary`.

AWS CLI

Para criar um vocabulário médico personalizado

O exemplo de `create-medical-vocabulary` a seguir cria um vocabulário personalizado. Para criar um vocabulário personalizado, você deve ter criado um arquivo de texto com todos os termos que deseja transcrever com mais precisão. Para `vocabulary-file-uri`, especifique o Amazon Simple Storage Service (Amazon URI S3) desse arquivo de texto. Em `language-code`, especifique um código de idioma correspondente ao idioma do vocabulário personalizado. Em `vocabulary-name`, especifique como você deseja chamar seu vocabulário personalizado.

```
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
```

Saída:

```
{
  "VocabularyName": "cli-medical-vocab-example",
  "LanguageCode": "language-code",
  "VocabularyState": "PENDING"
}
```

Para obter mais informações, consulte [Medical Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [CreateMedicalVocabulary](#) na Referência de AWS CLI Comandos.

create-vocabulary-filter

O código de exemplo a seguir mostra como usar `create-vocabulary-filter`.

AWS CLI

Para criar um filtro de vocabulário

O `create-vocabulary-filter` exemplo a seguir cria um filtro de vocabulário que usa um arquivo de texto que contém uma lista de palavras que você não gostaria que aparecessem em uma transcrição. Para código de idioma, especifique o código de idioma correspondente ao idioma do seu filtro de vocabulário. Para `vocabulary-filter-file-uri`, especifique o Amazon Simple Storage Service (Amazon URI S3) do arquivo de texto. Para `vocabulary-filter-name`, especifique o nome do seu filtro de vocabulário.

```
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
```

Saída:

```
{  
  "VocabularyFilterName": "cli-vocabulary-filter-example",  
  "LanguageCode": "language-code"  
}
```

Para obter mais informações, consulte [Filtragem de palavras indesejadas](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [CreateVocabularyFilter](#) na Referência de AWS CLI Comandos.

create-vocabulary

O código de exemplo a seguir mostra como usar `create-vocabulary`.

AWS CLI

Como criar um vocabulário personalizado

O exemplo de `create-vocabulary` a seguir cria um vocabulário personalizado. Para criar um vocabulário personalizado, você deve ter criado um arquivo de texto com todos os termos que deseja transcrever com mais precisão. Para `vocabulary-file-uri`, especifique o Amazon Simple Storage Service (Amazon URI S3) desse arquivo de texto. Em `language-code`, especifique um código de idioma correspondente ao idioma do vocabulário personalizado. Em `vocabulary-name`, especifique como você deseja chamar seu vocabulário personalizado.

```
aws transcribe create-vocabulary \  
  --language-code language-code \  
  --vocabulary-name cli-vocab-example \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/the-text-file-  
for-the-custom-vocabulary.txt
```

Saída:

```
{  
  "VocabularyName": "cli-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

Para obter mais informações, consulte [Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [CreateVocabulary](#) na Referência de AWS CLI Comandos.

delete-language-model

O código de exemplo a seguir mostra como usar delete-language-model.

AWS CLI

Para excluir um modelo de idioma personalizado

O delete-language-model exemplo a seguir exclui um modelo de linguagem personalizado.

```
aws transcribe delete-language-model \  
  --model-name model-name
```

Este comando não produz saída.

Para obter mais informações, consulte [Improving Domain-Specific Transcription Accuracy with Custom Language Models](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [DeleteLanguageModel](#) na Referência de AWS CLI Comandos.

delete-medical-transcription-job

O código de exemplo a seguir mostra como usar `delete-medical-transcription-job`.

AWS CLI

Como excluir um trabalho de transcrição médica

O exemplo de `delete-medical-transcription-job` a seguir exclui um trabalho de transcrição médica.

```
aws transcribe delete-medical-transcription-job \  
  --medical-transcription-job-name medical-transcription-job-name
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteMedicalTranscriptionJob](#) Amazon Transcribe Developer Guide.

- Para API obter detalhes, consulte [DeleteMedicalTranscriptionJob](#) na Referência de AWS CLI Comandos.

delete-medical-vocabulary

O código de exemplo a seguir mostra como usar `delete-medical-vocabulary`.

AWS CLI

Para excluir um vocabulário médico personalizado

O `delete-medical-vocabulary` exemplo a seguir exclui um vocabulário médico personalizado. Para nome do vocabulário, especifique o nome do vocabulário médico personalizado.

```
aws transcribe delete-vocabulary \  
  --vocabulary-name medical-custom-vocabulary-name
```

Este comando não produz saída.

Para obter mais informações, consulte [Medical Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [DeleteMedicalVocabulary](#) na Referência de AWS CLI Comandos.

delete-transcription-job

O código de exemplo a seguir mostra como usar delete-transcription-job.

AWS CLI

Como excluir um dos trabalhos de transcrição

O exemplo de delete-transcription-job a seguir exclui um dos trabalhos de transcrição.

```
aws transcribe delete-transcription-job \  
  --transcription-job-name your-transcription-job
```

Este comando não produz saída.

Para obter mais informações, consulte [DeleteTranscriptionJob](#) no Amazon Transcribe Developer Guide.

- Para API obter detalhes, consulte [DeleteTranscriptionJob](#) na Referência de AWS CLI Comandos.

delete-vocabulary-filter

O código de exemplo a seguir mostra como usar delete-vocabulary-filter.

AWS CLI

Para excluir um filtro de vocabulário

O delete-vocabulary-filter exemplo a seguir exclui um filtro de vocabulário.

```
aws transcribe delete-vocabulary-filter \  
  --vocabulary-filter-name vocabulary-filter-name
```

Este comando não produz saída.

Para obter mais informações, consulte [Filtragem de palavras indesejadas](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [DeleteVocabularyFilter](#) na Referência de AWS CLI Comandos.

delete-vocabulary

O código de exemplo a seguir mostra como usar delete-vocabulary.

AWS CLI

Como excluir um vocabulário personalizado

O exemplo de delete-vocabulary a seguir exclui um vocabulário personalizado.

```
aws transcribe delete-vocabulary \  
  --vocabulary-name vocabulary-name
```

Este comando não produz saída.

Para obter mais informações, consulte [Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [DeleteVocabulary](#) na Referência de AWS CLI Comandos.

describe-language-model

O código de exemplo a seguir mostra como usar describe-language-model.

AWS CLI

Para obter informações sobre um modelo de linguagem personalizado específico

O describe-language-model exemplo a seguir obtém informações sobre um modelo de linguagem personalizado específico. Por exemplo, abaixo, BaseModelName você pode ver se seu modelo é treinado usando um WideBand modelo NarrowBand ou. Modelos de linguagem personalizados com um modelo NarrowBand básico podem transcrever áudio com uma taxa de amostragem menor que 16. kHz Os modelos de linguagem que usam um modelo WideBand básico podem transcrever áudio com uma taxa de amostragem maior que 16. kHz O parâmetro S3Uri indica o prefixo Amazon S3 que você usou para acessar os dados de treinamento para criar o modelo de linguagem personalizado.

```
aws transcribe describe-language-model \  
  --base-model-name BaseModelName
```

```
--model-name cli-clm-example
```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Improving Domain-Specific Transcription Accuracy with Custom Language Models](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [DescribeLanguageModel](#) na Referência de AWS CLI Comandos.

get-medical-transcription-job

O código de exemplo a seguir mostra como usar `get-medical-transcription-job`.

AWS CLI

Para obter informações sobre um trabalho específico de transcrição médica

O `get-medical-transcription-job` exemplo a seguir obtém informações sobre um trabalho específico de transcrição médica. Para acessar os resultados da transcrição, use o `TranscriptFileUri` parâmetro. Se você ativou recursos adicionais para o trabalho de transcrição, poderá vê-los no objeto Configurações. O parâmetro `Especialidade` mostra a especialidade médica do provedor. O parâmetro `Tipo` indica se a fala na tarefa de transcrição é de uma conversa médica ou de um ditado médico.

```
aws transcribe get-medical-transcription-job \  
--medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Transcrição em lote no Guia](#) do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [GetMedicalTranscriptionJob](#) na Referência de AWS CLI Comandos.

get-medical-vocabulary

O código de exemplo a seguir mostra como usar get-medical-vocabulary.

AWS CLI

Para obter informações sobre um vocabulário médico personalizado

O `get-medical-vocabulary` exemplo a seguir obtém informações sobre um vocabulário médico personalizado. Você pode usar o `VocabularyState` parâmetro para ver o estado de processamento do vocabulário. Se `forREADY`, você pode usá-lo na `StartMedicalTranscriptionJob` operação. :

```
aws transcribe get-medical-vocabulary \  
  --vocabulary-name medical-vocab-example
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Medical Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [GetMedicalVocabulary](#) na Referência de AWS CLI Comandos.

get-transcription-job

O código de exemplo a seguir mostra como usar `get-transcription-job`.

AWS CLI

Como obter informações sobre um trabalho de transcrição específico

O exemplo de `get-transcription-job` a seguir coleta informações sobre um trabalho de transcrição específico. Para acessar os resultados da transcrição, use o `TranscriptFileUri` parâmetro. Use o `MediaFileUri` parâmetro para ver qual arquivo de áudio você transcreveu com

esse trabalho. É possível usar o objeto Configurações para ver os recursos opcionais habilitados no trabalho de transcrição.

```
aws transcribe get-transcription-job \  
--transcription-job-name your-transcription-job
```

Saída:

```
{  
  "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  
  }  
}
```

Para obter mais informações, consulte [Conceitos básicos \(interface de linha de AWS comando\)](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [GetTranscriptionJob](#) na Referência de AWS CLI Comandos.

get-vocabulary-filter

O código de exemplo a seguir mostra como usar `get-vocabulary-filter`.

AWS CLI

Para obter informações sobre um filtro de vocabulário

O `get-vocabulary-filter` exemplo a seguir obtém informações sobre um filtro de vocabulário. Você pode usar o `DownloadUri` parâmetro para obter a lista de palavras usadas para criar o filtro de vocabulário.

```
aws transcribe get-vocabulary-filter \  
  --vocabulary-filter-name testFilter
```

Saída:

```
{  
  "VocabularyFilterName": "testFilter",  
  "LanguageCode": "language-code",  
  "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",  
  "DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"  
}
```

Para obter mais informações, consulte [Filtrar palavras indesejadas](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [GetVocabularyFilter](#) na Referência de AWS CLI Comandos.

get-vocabulary

O código de exemplo a seguir mostra como usar `get-vocabulary`.

AWS CLI

Como obter informações sobre um vocabulário personalizado

O exemplo de `get-vocabulary` a seguir coleta informações sobre um vocabulário personalizado criado anteriormente.

```
aws transcribe get-vocabulary \  
  --vocabulary-name cli-vocab-1
```

Saída:

```
{
```

```
"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"
}
```

Para obter mais informações, consulte [Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [GetVocabulary](#) na Referência de AWS CLI Comandos.

list-language-models

O código de exemplo a seguir mostra como usar `list-language-models`.

AWS CLI

Para listar seus modelos de linguagem personalizados

O `list-language-models` exemplo a seguir lista os modelos de idioma personalizados associados à sua AWS conta e região. Você pode usar os `TuningDataS3Uri` parâmetros `S3Uri` e para encontrar os prefixos do Amazon S3 que você usou como dados de treinamento ou dados de ajuste. `BaseModelName` informa se você usou um `NarrowBand` `WideBand` modelo ou para criar um modelo de linguagem personalizado. Você pode transcrever áudio com uma taxa de amostragem inferior a 16 kHz com um modelo de idioma personalizado usando um modelo `NarrowBand` básico. Você pode transcrever áudio 16 kHz ou superior com um modelo de idioma personalizado usando um modelo `WideBand` básico. O `ModelStatus` parâmetro mostra se você pode usar o modelo de idioma personalizado em um trabalho de transcrição. Se o valor for `COMPLETED`, você poderá usá-lo em um trabalho de transcrição.

```
aws transcribe list-language-models
```

Saída:

```
{
  "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"
    }
  }
]
```


Para obter mais informações, consulte [Improving Domain-Specific Transcription Accuracy with Custom Language Models](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [ListLanguageModels](#) na Referência de AWS CLI Comandos.

list-medical-transcription-jobs

O código de exemplo a seguir mostra como usar `list-medical-transcription-jobs`.

AWS CLI

Como listar os trabalhos de transcrição médica

O `list-medical-transcription-jobs` exemplo a seguir lista os trabalhos de transcrição médica associados à sua AWS conta e região. Para obter mais informações sobre um trabalho de transcrição específico, copie o valor de um `MedicalTranscriptionJobName` parâmetro na saída da transcrição e especifique esse valor para a `MedicalTranscriptionJobName` opção do `get-medical-transcription-job` comando. Para ver mais trabalhos de transcrição, copie o valor do `NextToken` parâmetro, execute o `list-medical-transcription-jobs` comando novamente e especifique esse valor na `--next-token` opção.

```
aws transcribe list-medical-transcription-jobs
```

Saída:

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb0l
+LaDtZPC4u6ttoHLR1EfzqstHXSgapXg3tEBtm9piIaPB6M0M5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPBQ0HPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSenzq49Q9xHgHdT2yBADFmvFK7myZBs75+2vQZ0SVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgsJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYox3tTPW1D7phpbVSYkrh/
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"
  }
]
```

Para obter mais informações, consulte <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html> no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [ListMedicalTranscriptionJobs](#) na Referência de AWS CLI Comandos.

list-medical-vocabularies

O código de exemplo a seguir mostra como usar `list-medical-vocabularies`.

AWS CLI

Para listar seus vocabulários médicos personalizados

O `list-medical-vocabularies` exemplo a seguir lista os vocabulários médicos personalizados associados à sua AWS conta e região. Para obter mais informações sobre um trabalho de transcrição específico, copie o valor de um `MedicalTranscriptionJobName` parâmetro na saída da transcrição e especifique esse valor para a `MedicalTranscriptionJobName` opção do `get-medical-transcription-job` comando. Para ver mais trabalhos de transcrição, copie o valor do `NextToken` parâmetro, execute o `list-medical-transcription-jobs` comando novamente e especifique esse valor na `--next-token` opção.

```
aws transcribe list-medical-vocabularies
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Medical Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [ListMedicalVocabularies](#) na Referência de AWS CLI Comandos.

list-transcription-jobs

O código de exemplo a seguir mostra como usar `list-transcription-jobs`.

AWS CLI

Como listar os trabalhos de transcrição

O `list-transcription-jobs` exemplo a seguir lista os trabalhos de transcrição associados à sua AWS conta e região.

```
aws transcribe list-transcription-jobs
```

Saída:

```
{
  "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"
  }
]
```

Para obter mais informações, consulte [Conceitos básicos \(interface de linha de AWS comando\)](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [ListTranscriptionJobs](#) na Referência de AWS CLI Comandos.

list-vocabularies

O código de exemplo a seguir mostra como usar `list-vocabularies`.

AWS CLI

Como listar os vocabulários personalizados

O `list-vocabularies` exemplo a seguir lista os vocabulários personalizados associados à sua AWS conta e região.

```
aws transcribe list-vocabularies
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações, consulte [Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [ListVocabularies](#) na Referência de AWS CLI Comandos.

list-vocabulary-filters

O código de exemplo a seguir mostra como usar `list-vocabulary-filters`.

AWS CLI

Para listar seus filtros de vocabulário

O `list-vocabulary-filters` exemplo a seguir lista os filtros de vocabulário associados à sua AWS conta e região.

```
aws transcribe list-vocabulary-filters
```

Saída:

```
{
  "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"
}
]
```

Para obter mais informações, consulte [Filtragem de palavras indesejadas](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [ListVocabularyFilters](#) na Referência de AWS CLI Comandos.

start-medical-transcription-job

O código de exemplo a seguir mostra como usar `start-medical-transcription-job`.

AWS CLI

Exemplo 1: como transcrever um ditado médico armazenado como um arquivo de áudio

O exemplo de `start-medical-transcription-job` a seguir transcreve um arquivo de áudio. Especifique o local da saída da transcrição no parâmetro `OutputBucketName`.


```
aws transcribe start-medical-transcription-job \  
--cli-input-json file://myfile.json
```

Conteúdo de `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"  
  }  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Batch Transcription Overview](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 2: como transcrever um diálogo entre médico e paciente armazenado como um arquivo de áudio

O exemplo de `start-medical-transcription-job` a seguir transcreve um arquivo de áudio contendo um diálogo entre médico e paciente. Você especifica a localização da saída da transcrição no `OutputBucketName` parâmetro.

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://mysecondfile.json
```

Conteúdo de `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"  
  }  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Batch Transcription Overview](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 3: como transcrever um arquivo de áudio multicanal de um diálogo entre médico e paciente

O exemplo de `start-medical-transcription-job` a seguir transcreve o áudio de cada canal no arquivo de áudio e mescla as transcrições separadas de cada canal em uma única saída de transcrição. Especifique o local da saída da transcrição no parâmetro `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://mythirdfile.json
```

Conteúdo de `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  
  }  
}
```

Saída:

```
{  
  "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"
  }
}
```

Para obter mais informações, consulte [Channel Identification](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 4: como transcrever um arquivo de áudio de um diálogo entre médico e paciente e identificar quem está falando na saída da transcrição

O exemplo de `start-medical-transcription-job` a seguir transcreve um arquivo de áudio e rotula a fala de cada locutor na saída da transcrição. Especifique o local da saída da transcrição no parâmetro `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfourthfile.json
```

Conteúdo de `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
  }
}
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Identifying Speakers](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 5: como transcrever uma conversa médica armazenada como um arquivo de áudio com até duas alternativas de transcrição

O exemplo de `start-medical-transcription-job` a seguir cria até duas transcrições alternativas de um único arquivo de áudio. Cada transcrição tem um nível de confiança associado a ela. Por padrão, o Amazon Transcribe retorna a transcrição com o maior nível de confiança. É possível especificar que o Amazon Transcribe retorne transcrições adicionais com níveis de confiança mais baixos. Especifique o local da saída da transcrição no parâmetro `OutputBucketName`.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfifthfile.json
```

Conteúdo de `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
}
}
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Alternative Transcriptions](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 6: como transcrever um arquivo de áudio de um ditado médico com até duas transcrições alternativas

O exemplo de `start-medical-transcription-job` a seguir transcreve um arquivo de áudio e usa um filtro de vocabulário para mascarar qualquer palavra indesejada. Você especifica a localização da saída da transcrição no `OutputBucketName` parâmetro.

```
aws transcribe start-medical-transcription-job \  
--cli-input-json file://mysixthfile.json
```

Conteúdo de `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  
  }  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Alternative Transcriptions](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 7: como transcrever um arquivo de áudio de um ditado médico com maior precisão usando um vocabulário personalizado

O exemplo de `start-medical-transcription-job` a seguir transcreve um arquivo de áudio e usa um vocabulário médico personalizado que você criou anteriormente para aumentar a precisão da transcrição. Especifique o local da saída da transcrição no parâmetro `OutputBucketName`.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myseventhfile.json
```

Conteúdo de `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"  
  }  
}
```

Saída:

```
{  
  "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"
  }
}

```

Para obter mais informações, consulte [Medical Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [StartMedicalTranscriptionJob](#) na Referência de AWS CLI Comandos.

start-transcription-job

O código de exemplo a seguir mostra como usar `start-transcription-job`.

AWS CLI

Exemplo 1: como transcrever um arquivo de áudio

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myfile.json

```

Conteúdo de `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"
  }
}

```

Para obter mais informações, consulte [Conceitos básicos \(interface de linha de AWS comando\)](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 2: como transcrever um arquivo de áudio multicanal

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio multicanal.

```
aws transcribe start-transcription-job \  
--cli-input-json file://mysecondfile.json
```

Conteúdo de `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  
  }  
}
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

Para obter mais informações, consulte [Transcribing Multi-Channel Audio](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 3: como transcrever um arquivo de áudio e identificar quem está falando

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio e identifica os locutores na saída da transcrição.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mythirdfile.json
```

Conteúdo de `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  
  }  
}
```

Saída:

```
{  
  "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  
    }  
  }  
}
```

Para obter mais informações, consulte [Identifying Speakers](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 4: como transcrever um arquivo de áudio e mascarar palavras indesejadas na saída da transcrição

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio e usa um filtro de vocabulário que você criou anteriormente para mascarar palavras indesejadas.

```
aws transcribe start-transcription-job \  
--cli-input-json file://myfourthfile.json
```

Conteúdo de `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"  
  }  
}
```

Saída:

```
{  
  "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"  
    }  
  }  
}
```

```

    }
  }
}

```

Para obter mais informações, consulte [Filtering Transcriptions](#) o Guia do desenvolvedor do Amazon Transcribe.

Exemplo 5: como transcrever um arquivo de áudio e remover palavras indesejadas na saída da transcrição

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio e usa um filtro de vocabulário que você criou anteriormente para mascarar palavras indesejadas.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myfifthfile.json

```

Conteúdo de `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"
  }
}

```

Saída:

```

{
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Filtering Transcriptions](#) o Guia do desenvolvedor do Amazon Transcribe.

Exemplo 6: como transcrever um arquivo de áudio com maior precisão usando um vocabulário personalizado

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio e usa um filtro de vocabulário que você criou anteriormente para mascarar palavras indesejadas.

```

aws transcribe start-transcription-job \
  --cli-input-json file://mysixthfile.json

```

Conteúdo de `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"
  }
}

```

Saída:

```

{
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Filtering Transcriptions](#) o Guia do desenvolvedor do Amazon Transcribe.

Exemplo 7: como identificar o idioma de um arquivo de áudio e transcrevê-lo

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio e usa um filtro de vocabulário que você criou anteriormente para mascarar palavras indesejadas.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json

```

Conteúdo de `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"
  }
}

```

Saída:

```

{
  "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
  }
}
```

Para obter mais informações, consulte [Identifying the Language](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 8: como transcrever um arquivo de áudio com informações de identificação pessoal censuradas

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio e oculta qualquer informação de identificação pessoal na saída da transcrição.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myeighthfile.json
```

Conteúdo de `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"
  }
}
```

Saída:

```
{
  "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"
    }
  }
}

```

Para obter mais informações, consulte [Automatic Content Redaction](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 9: Para gerar uma transcrição com informações de identificação pessoal (PII) editadas e uma transcrição não editada

O exemplo de `start-transcription-job` a seguir gera duas transcrições do arquivo de áudio, uma com as informações de identificação pessoal editadas e a outra sem nenhuma edição.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myninthfile.json

```

Conteúdo de `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"
  }
}

```

Saída:

```

{

```

```

"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"
  }
}
}

```

Para obter mais informações, consulte [Automatic Content Redaction](#) no Guia do desenvolvedor do Amazon Transcribe.

Exemplo 10: como usar um modelo de linguagem personalizado criado anteriormente para transcrever um arquivo de áudio.

O exemplo de `start-transcription-job` a seguir transcreve o arquivo de áudio com um modelo de idioma personalizado que você criou anteriormente.

```

aws transcribe start-transcription-job \
  --cli-input-json file://mytenthfile.json

```

Conteúdo de `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"
  }
}

```

Saída:

```
{
  "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"
    }
  }
}
```

Para obter mais informações, consulte [Improving Domain-Specific Transcription Accuracy with Custom Language Models](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [StartTranscriptionJob](#) na Referência de AWS CLI Comandos.

update-medical-vocabulary

O código de exemplo a seguir mostra como usar `update-medical-vocabulary`.

AWS CLI

Para atualizar um vocabulário médico personalizado com novos termos.

O `update-medical-vocabulary` exemplo a seguir substitui os termos usados em um vocabulário médico personalizado por novos. Pré-requisito: para substituir os termos em um vocabulário médico personalizado, você precisa de um arquivo com novos termos.

```
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
```

Saída:

```
{
```

```
"VocabularyName": "medical-custom-vocabulary",
"LanguageCode": "en-US",
"VocabularyState": "PENDING"
}
```

Para obter mais informações, consulte [Medical Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [UpdateMedicalVocabulary](#) na Referência de AWS CLI Comandos.

update-vocabulary-filter

O código de exemplo a seguir mostra como usar `update-vocabulary-filter`.

AWS CLI

Para substituir as palavras em um filtro de vocabulário

O `update-vocabulary-filter` exemplo a seguir substitui as palavras em um filtro de vocabulário por novas. Pré-requisito: Para atualizar um filtro de vocabulário com as novas palavras, você deve salvá-las como um arquivo de texto.

```
aws transcribe update-vocabulary-filter \
  --vocabulary-filter-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/your-text-
  file-to-update-your-vocabulary-filter.txt \
  --vocabulary-filter-name vocabulary-filter-name
```

Saída:

```
{
  "VocabularyFilterName": "vocabulary-filter-name",
  "LanguageCode": "language-code",
  "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"
}
```

Para obter mais informações, consulte [Filtragem de palavras indesejadas](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [UpdateVocabularyFilter](#) na Referência de AWS CLI Comandos.

update-vocabulary

O código de exemplo a seguir mostra como usar `update-vocabulary`.

AWS CLI

Como atualizar um vocabulário personalizado com novos termos.

O exemplo de `update-vocabulary` a seguir substitui os termos usados para criar o vocabulário personalizado pelos novos termos fornecidos. Pré-requisito: para substituir os termos em um vocabulário personalizado, você precisa de um arquivo com novos termos.

```
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
```

Saída:

```
{  
  "VocabularyName": "custom-vocabulary",  
  "LanguageCode": "language",  
  "VocabularyState": "PENDING"  
}
```

Para obter mais informações, consulte [Custom Vocabularies](#) no Guia do desenvolvedor do Amazon Transcribe.

- Para API obter detalhes, consulte [UpdateVocabulary](#) na Referência de AWS CLI Comandos.

Exemplos do Amazon Translate usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com o Amazon Translate.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

import-terminology

O código de exemplo a seguir mostra como usar `import-terminology`.

AWS CLI

Para importar uma terminologia personalizada de um arquivo

O `import-terminology` exemplo a seguir cria uma terminologia chamada a `MyTestTerminology` partir do `test-terminology.csv` arquivo:

```
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
```

Conteúdo de `test-terminology.csv`:

en, fr, es, zh Olá, mundo! , Bonjour tout le monde! , Olá Mundo! ,???? Amazon, Amazônia, Amazônia, Amazônia

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [ImportTerminology](#) na Referência de AWS CLI Comandos.

Trusted Advisor exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with Trusted Advisor.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-organization-recommendation

O código de exemplo a seguir mostra como usar `get-organization-recommendation`.

AWS CLI

Para obter uma recomendação de organização

O `get-organization-recommendation` exemplo a seguir obtém uma recomendação de organização por meio de seu identificador.

```
aws trustedadvisor get-organization-recommendation \  
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-  
recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [GetOrganizationRecommendation](#) na Referência de AWS CLI Comandos.

get-recommendation

O código de exemplo a seguir mostra como usar `get-recommendation`.

AWS CLI

Para obter uma recomendação

O `get-recommendation` exemplo a seguir recebe uma recomendação por meio de seu identificador.

```
aws trustedadvisor get-recommendation \  
  --recommendation-  
  identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-  
  bbb7-491a-833b-5773e9589578
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [GetRecommendation](#) na Referência de AWS CLI Comandos.

list-checks

O código de exemplo a seguir mostra como usar `list-checks`.

AWS CLI

Para listar as verificações do Trusted Advisor

O `list-checks` exemplo a seguir lista todas as verificações do Trusted Advisor.

```
aws trustedadvisor list-checks
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [ListChecks](#) na Referência de AWS CLI Comandos.

list-organization-recommendation-accounts

O código de exemplo a seguir mostra como usar `list-organization-recommendation-accounts`.

AWS CLI

Para listar contas de recomendação da organização

O `list-organization-recommendation-accounts` exemplo a seguir lista todos os resumos de recomendações de contas para uma recomendação de organização por meio de seu identificador.

```
aws trustedadvisor list-organization-recommendation-accounts \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-
  recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [ListOrganizationRecommendationAccounts](#) na Referência de AWS CLI Comandos.

list-organization-recommendation-resources

O código de exemplo a seguir mostra como usar `list-organization-recommendation-resources`.

AWS CLI

Para listar os recursos de recomendação da organização

O `list-organization-recommendation-resources` exemplo a seguir lista todos os recursos para uma recomendação de organização por seu identificador.

```
aws trustedadvisor list-organization-recommendation-resources \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

Saída:

```
{
  "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/
f4d01bd20f4cd5372062aaafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
      "awsResourceId": "database-2-instance-1-us-west-2a",
      "id":
"f4d01bd20f4cd5372062aaafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
      "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"
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [ListOrganizationRecommendationResources](#) na Referência de AWS CLI Comandos.

list-organization-recommendations

O código de exemplo a seguir mostra como usar `list-organization-recommendations`.

AWS CLI

Exemplo 1: Para listar as recomendações da organização

O `list-organization-recommendations` exemplo a seguir lista todas as recomendações da organização e não inclui um filtro.

```
aws trustedadvisor list-organization-recommendations
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

Exemplo 2: Para listar as recomendações da organização com um filtro

O `list-organization-recommendations` exemplo a seguir filtra e retorna no máximo uma recomendação da organização que faz parte do pilar “segurança”.

```
aws trustedadvisor list-organization-recommendations \  
  --pillar security \  
  --max-items 100
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

Exemplo 3: Para listar as recomendações da organização com um token de paginação

O `list-organization-recommendations` exemplo a seguir usa o "nextToken" retornado de uma solicitação anterior para buscar a próxima página de recomendações da organização.

```
aws trustedadvisor list-organization-recommendations \  
  --pillar security \  
  --max-items 100
```

```
--max-items 100 \  
--starting-token <next-token>
```

Saída:

```
{  
  "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"  
  }]  
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [ListOrganizationRecommendations](#) na Referência de AWS CLI Comandos.

list-recommendation-resources

O código de exemplo a seguir mostra como usar `list-recommendation-resources`.

AWS CLI

Para listar recursos de recomendação

O `list-recommendation-resources` exemplo a seguir lista todos os recursos de uma recomendação por seu identificador.

```
aws trustedadvisor list-recommendation-resources \
  --recommendation-
  identifíer arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
  bbb7-491a-833b-5773e9589578
```

Saída:

```
{
  "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/
      e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefce9eb63e",
      "id":
      "e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefce9eb63e",
      "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"
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [ListRecommendationResources](#) na Referência de AWS CLI Comandos.

list-recommendations

O código de exemplo a seguir mostra como usar `list-recommendations`.

AWS CLI

Exemplo 1: Para listar recomendações

O `list-recommendations` exemplo a seguir lista todas as recomendações e não inclui um filtro.

```
aws trustedadvisor list-recommendations
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

Exemplo 2: Para listar recomendações com um filtro

O `list-recommendations` exemplo a seguir lista as recomendações e inclui um filtro.

```
aws trustedadvisor list-recommendations \  
  --aws-service iam \  
  --max-items 100
```

Saída:

```
{  
  "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"
```



```
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

Exemplo 3: Para listar recomendações com um token de paginação

O `list-recommendations` exemplo a seguir usa o "nextToken" retornado de uma solicitação anterior para buscar a próxima página de recomendações filtradas.

```
aws trustedadvisor list-recommendations \  
  --aws-service rds \  
  --max-items 100 \  
  --starting-token <next-token>
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [ListRecommendations](#) na Referência de AWS CLI Comandos.

update-organization-recommendation-lifecycle

O código de exemplo a seguir mostra como usar `update-organization-recommendation-lifecycle`.

AWS CLI

Para atualizar o ciclo de vida das recomendações de uma organização

O `update-organization-recommendation-lifecycle` exemplo a seguir atualiza o ciclo de vida de uma recomendação da organização por meio de seu identificador.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [UpdateOrganizationRecommendationLifecycle](#) na Referência de AWS CLI Comandos.

update-recommendation-lifecycle

O código de exemplo a seguir mostra como usar `update-recommendation-lifecycle`.

AWS CLI

Para atualizar o ciclo de vida de uma recomendação

O `update-recommendation-lifecycle` exemplo a seguir atualiza o ciclo de vida de uma recomendação por meio de seu identificador.

```
aws trustedadvisor update-recommendation-lifecycle \  
  --recommendation-  
  identifiier arn:aws:trustedadvisor::000000000000:recommendation/861c9c6e-  
  f169-405a-8b59-537a8cacc7a \  
  --lifecycle-stage resolved \  
  --update-reason-code valid_business_case
```

Este comando não produz saída.

Para obter mais informações, consulte [Get started with the Trusted Advisor API](#) no AWS Trusted Advisor User Guide.

- Para API obter detalhes, consulte [UpdateRecommendationLifecycle](#) na Referência de AWS CLI Comandos.

Exemplos de permissões verificadas usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com permissões verificadas.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-identity-source

O código de exemplo a seguir mostra como usar `create-identity-source`.

AWS CLI

Para criar uma fonte de identidade

O `create-identity-source` exemplo a seguir cria uma fonte de identidade que permite referenciar identidades armazenadas no grupo de usuários do Amazon Cognito especificado. Essas identidades estão disponíveis nas Permissões verificadas como entidades do tipo `User`.

```
aws verifiedpermissions create-identity-source \
  --configuration file://config.txt \
  --principal-entity-type "User" \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Conteúdo de `config.txt`:

```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5",
    "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
  }
}
```

Saída:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

Para obter mais informações sobre fontes de identidade, consulte [Uso de permissões verificadas da Amazon com provedores de identidade](#) no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [CreateIdentitySource](#) na Referência de AWS CLI Comandos.

create-policy-store

O código de exemplo a seguir mostra como usar create-policy-store.

AWS CLI

Para criar um repositório de políticas

O create-policy-store exemplo a seguir cria um repositório de políticas na AWS região atual.

```
aws verifiedpermissions create-policy-store \  
--validation-settings "mode=STRICT"
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre repositórios de políticas, consulte Armazenamentos de [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [CreatePolicyStore](#) na Referência de AWS CLI Comandos.

create-policy-template

O código de exemplo a seguir mostra como usar create-policy-template.

AWS CLI

Exemplo 1: Para criar um modelo de política

O create-policy-template exemplo a seguir cria um modelo de política com uma declaração que contém um espaço reservado para o diretor.

```
aws verifiedpermissions create-policy-template \  
  --definition file://template1.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Conteúdo do arquivo `template1.txt`:

```
permit(  
  principal in ?principal,  
  action == Action::"view",  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Saída:

```
{  
  "createdDate": "2023-06-12T20:47:42.804511+00:00",  
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyTemplateId": "PTEXAMPLEabcdefg111111"  
}
```

Para obter mais informações sobre modelos de políticas, consulte Modelos de [política de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [CreatePolicyTemplate](#) na Referência de AWS CLI Comandos.

create-policy

O código de exemplo a seguir mostra como usar `create-policy`.

AWS CLI

Exemplo 1: Para criar uma política estática

O `create-policy` exemplo a seguir cria uma política estática com um escopo de política que especifica tanto um principal quanto um recurso.

```
aws verifiedpermissions create-policy \  
  --definition file://definition1.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Conteúdo do arquivo `definition1.txt`:

```
{
  "static": {
    "description": "Grant everyone of janeFriends UserGroup access to the
vacationFolder Album",
    "statement": "permit(principal in UserGroup::\"janeFriends\", action,
resource in Album::\"vacationFolder\" );"
  }
}
```

Saída:

```
{
  "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"
  }
}
```

Exemplo 2: Para criar uma política estática que conceda acesso a um recurso para todos

O `create-policy` exemplo a seguir cria uma política estática com um escopo de política que especifica somente um recurso.

```
aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEabcdefg111111
```

Conteúdo do arquivo `definition2.txt`:

```
{
```

```

    "static": {
      "description": "Grant everyone access to the publicFolder Album",
      "statement": "permit(principal, action, resource in Album:\""publicFolder
\");"
    }
  }
}

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 3: Para criar uma política vinculada ao modelo que esteja associada ao modelo especificado

O `create-policy` exemplo a seguir cria uma política vinculada ao modelo usando o modelo de política especificado e associa o principal especificado a ser usado à nova política vinculada ao modelo.

```

aws verifiedpermissions create-policy \
  --definition file://definition.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111

```

Conteúdo de `definition.txt`:

```

{
  "templateLinked": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    }
  }
}

```



```
}  
}
```

Saída:

```
{  
  "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"  
  }  
}
```

Para obter mais informações sobre políticas, consulte as [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [CreatePolicy](#) na Referência de AWS CLI Comandos.

delete-identity-source

O código de exemplo a seguir mostra como usar delete-identity-source.

AWS CLI

Para excluir uma fonte de identidade

O delete-identity-source exemplo a seguir exclui a fonte de identidade que tem o ID especificado.

```
aws verifiedpermissions delete-identity-source \  
  --identity-source-id ISEXAMPLEabcdefg111111 \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Este comando não produz saída.


```
--policy-store-id PSEXAMPLEEabcdefg111111
```

Este comando não produz saída.

Para obter mais informações sobre modelos de políticas, consulte Modelos de [política de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [DeletePolicyTemplate](#) na Referência de AWS CLI Comandos.

delete-policy

O código de exemplo a seguir mostra como usar delete-policy.

AWS CLI

Para excluir uma política estática ou vinculada a um modelo

O delete-policy exemplo a seguir exclui a política que tem o ID especificado.

```
aws verifiedpermissions delete-policy \  
  --policy-id SPEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Este comando não produz saída.

Para obter mais informações sobre políticas, consulte as [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

get-identity-source

O código de exemplo a seguir mostra como usar get-identity-source.

AWS CLI

Para recuperar detalhes sobre uma fonte de identidade

O get-identity-source exemplo a seguir exibe os detalhes da fonte de identidade com o ID especificado.

```
aws verifiedpermissions get-identity-source \  
  --policy-id SPEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

```
--identity-source ISEXAMPLEEabcdefg111111 \  
--policy-store-id PSEXAMPLEEabcdefg111111
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre fontes de identidade, consulte [Uso de permissões verificadas da Amazon com provedores de identidade](#) no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [GetIdentitySource](#) na Referência de AWS CLI Comandos.

get-policy-store

O código de exemplo a seguir mostra como usar `get-policy-store`.

AWS CLI

Para recuperar detalhes sobre um repositório de políticas

O `get-policy-store` exemplo a seguir exibe os detalhes do repositório de políticas com o ID especificado.

```
aws verifiedpermissions get-policy-store \  
--policy-store-id PSEXAMPLEEabcdefg111111
```

Saída:

```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
  "createdDate": "2023-06-05T20:16:46.225598+00:00",
  "lastUpdatedDate": "2023-06-08T20:40:23.173691+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "validationSettings": { "mode": "OFF" }
}
```

Para obter mais informações sobre repositórios de políticas, consulte Armazenamentos de [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [GetPolicyStore](#) na Referência de AWS CLI Comandos.

get-policy-template

O código de exemplo a seguir mostra como usar `get-policy-template`.

AWS CLI

Para recuperar detalhes sobre um modelo de política

O `get-policy-template` exemplo a seguir exibe os detalhes do modelo de política com a ID especificada.

```
aws verifiedpermissions get-policy-template \
  --policy-template-id PTEXAMPLEabcdefg111111 \
  --policy-store-id PSEXAMPLEabcdefg111111
```

Saída:

```
{
  "createdDate": "2023-06-12T20:47:42.804511+00:00",
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEabcdefg111111",
  "statement": "permit(\n  principal in ?principal,\n  action == Action::
\nview\", \n  resource == Photo::\"VacationPhoto94.jpg\" \n);\"
}
```

Para obter mais informações sobre modelos de políticas, consulte Modelos de [política de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [GetPolicyTemplate](#) na Referência de AWS CLI Comandos.

get-policy

O código de exemplo a seguir mostra como usar `get-policy`.

AWS CLI

Para recuperar detalhes sobre uma política

O `get-policy` exemplo a seguir exibe os detalhes da política com o ID especificado.

```
aws verifiedpermissions get-policy \  
  --policy-id PSEXAMPLEabcdefg111111 \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Saída:

```
{  
  "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"  
  }  
}
```

```
}
```

Para obter mais informações sobre políticas, consulte as [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [GetPolicy](#) na Referência de AWS CLI Comandos.

get-schema

O código de exemplo a seguir mostra como usar `get-schema`.

AWS CLI

Para recuperar o esquema em um repositório de políticas

O `get-schema` exemplo a seguir exibe os detalhes do esquema no repositório de políticas especificado.

```
aws verifiedpermissions get-schema \  
--policy-store-id PSEXAMPLEabcdefg111111
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações sobre o esquema, consulte Esquema do [repositório de políticas no Guia](#) do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [GetSchema](#) na Referência de AWS CLI Comandos.

is-authorized-with-token

O código de exemplo a seguir mostra como usar `is-authorized-with-token`.

AWS CLI

Exemplo 1: Para solicitar uma decisão de autorização para uma solicitação de usuário (permitir)

O `is-authorized-with-token` exemplo a seguir solicita uma decisão de autorização para um usuário que foi autenticado pelo Amazon Cognito. A solicitação usa o token de identidade fornecido pelo Cognito em vez do token de acesso. Neste exemplo, o armazenamento de informações especificado está configurado para retornar entidades principais como entidades do 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"
```

O repositório de políticas contém uma política com a seguinte declaração que aceita identidades do grupo de usuários e ID do aplicativo do Cognito especificados.

```
permit(  
  principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0ka1bmc",  
  action,  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Saída:

```
{  
  "decision":"Allow",  
  "determiningPolicies":[  
    {  
      "determiningPolicyId":"SPEXAMPLEabcdefg111111"  
    }  
  ],  
  "errors":[]  
}
```

Para obter mais informações sobre o uso de identidades de um grupo de usuários do Cognito, consulte [Usando as Permissões Verificadas da Amazon com provedores de identidade](#) no Guia do Usuário de Permissões Verificadas da Amazon.

- Para API obter detalhes, consulte [IsAuthorizedWithToken](#) na Referência de AWS CLI Comandos.

is-authorized

O código de exemplo a seguir mostra como usar `is-authorized`.

AWS CLI

Exemplo 1: Para solicitar uma decisão de autorização para uma solicitação de usuário (permitir)

O `is-authorized` exemplo a seguir solicita uma decisão de autorização para um principal do tipo `User` chamado `Alice`, que deseja realizar a `updatePhoto` operação, em um recurso do tipo `Photo` chamado `VacationPhoto94.jpg`.

A resposta mostra que a solicitação é permitida por uma política.

```
aws verifiedpermissions is-authorized \  
  --principal entityType=User,entityId=alice \  
  --action actionType=Action,actionId=view \  
  --resource entityType=Photo,entityId=VactionPhoto94.jpg \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Saída:

```
{  
  "decision": "ALLOW",  
  "determiningPolicies": [  
    {  
      "policyId": "SPEXAMPLEabcdefg111111"  
    }  
  ],  
  "errors": []  
}
```

Exemplo 2: Para solicitar uma decisão de autorização para uma solicitação de usuário (negar)

O exemplo a seguir é igual ao exemplo anterior, exceto que o principal é `User::"Bob"`. O repositório de políticas não contém nenhuma política que permita que esse usuário acesse `Album::"alice_folder"` o.

A saída indica que `Deny` estava implícito porque a lista de `DeterminingPolicies` está vazia.

```
aws verifiedpermissions create-policy \  
  --definition file://definition2.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Saída:

```
{  
  "decision": "DENY",  
  "determiningPolicies": [],  
  "errors": []  
}
```

Para obter mais informações, consulte o [Guia do usuário de permissões verificadas da Amazon](#).

- Para API obter detalhes, consulte [IsAuthorized](#) na Referência de AWS CLI Comandos.

list-identity-sources

O código de exemplo a seguir mostra como usar `list-identity-sources`.

AWS CLI

Para listar as fontes de identidade disponíveis

O `list-identity-sources` exemplo a seguir lista todas as fontes de identidade no repositório de políticas especificado.

```
aws verifiedpermissions list-identity-sources \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Saída:

```
{  
  "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": "ISEXAMPLEabcdefgh111111",
      "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
      "policyStoreId": "PSEXAMPLEabcdefgh111111",
      "principalEntityType": "User"
    }
  ]
}
```

Para obter mais informações sobre fontes de identidade, consulte [Uso de permissões verificadas da Amazon com provedores de identidade](#) no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [ListIdentitySources](#) na Referência de AWS CLI Comandos.

list-policies

O código de exemplo a seguir mostra como usar `list-policies`.

AWS CLI

Para listar as políticas disponíveis

O `list-policies` exemplo a seguir lista todas as políticas no repositório de políticas especificado.

```
aws verifiedpermissions list-policies \
  --policy-store-id PSEXAMPLEabcdefgh111111
```

Saída:

```
{
  "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"
        }
    ]
}
```

Para obter mais informações sobre políticas, consulte as [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [ListPolicies](#) na Referência de AWS CLI Comandos.

list-policy-stores

O código de exemplo a seguir mostra como usar `list-policy-stores`.

AWS CLI

Para listar os repositórios de políticas disponíveis

O `list-policy-stores` exemplo a seguir lista todos os repositórios de políticas na AWS região. Todos os comandos para permissões verificadas, exceto `create-policy-store` e `list-policy-stores` exigem que você especifique o ID do repositório de políticas com o qual deseja trabalhar.

```
aws verifiedpermissions list-policy-stores
```

Saída:

```
{
  "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": "PSEXAMPLEEabcdefg222222"
  },
  {
    "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEEabcdefg333333",
    "createdDate": "2023-06-08T18:09:46.920600+00:00",
    "policyStoreId": "PSEXAMPLEEabcdefg333333"
  }
]
}
```

Para obter mais informações sobre repositórios de políticas, consulte Armazenamentos de [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [ListPolicyStores](#) na Referência de AWS CLI Comandos.

list-policy-templates

O código de exemplo a seguir mostra como usar `list-policy-templates`.

AWS CLI

Para listar os modelos de política disponíveis

O `list-policy-templates` exemplo a seguir lista todos os modelos de política no repositório de políticas especificado.

```
aws verifiedpermissions list-policy-templates \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Saída:

```
{
  "policyTemplates": [
    {
      "createdDate": "2023-06-12T20:47:42.804511+00:00",
      "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
      "policyStoreId": "PSEXAMPLEEabcdefg111111",
      "policyTemplateId": "PTEXAMPLEEabcdefg111111"
    }
  ]
}
```

```

    ]
  }
}

```

Para obter mais informações sobre modelos de políticas, consulte Modelos de [política de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [ListPolicyTemplates](#) na Referência de AWS CLI Comandos.

put - schema

O código de exemplo a seguir mostra como usar put - schema.

AWS CLI

Para salvar um esquema em um repositório de políticas

O put - schema exemplo a seguir cria ou substitui o esquema no repositório de políticas especificado.

O cedarJson parâmetro no arquivo de entrada usa uma representação em cadeia de caracteres de um JSON objeto. Ele contém aspas incorporadas (") dentro do par de aspas mais externo. Isso exige que você converta o JSON em uma string precedendo todas as aspas incorporadas com um caractere de barra invertida (\) e combinando todas as linhas em uma única linha de texto sem quebras de linha.

As sequências de caracteres de exemplo podem ser exibidas em várias linhas aqui para facilitar a leitura, mas a operação exige que os parâmetros sejam enviados como cadeias de caracteres de uma única linha.

```

aws iam put-permission-verification-policy --definition file: //schema.txt -- policy-store-id
PSEXAMPLEabcdefg111111

```

Conteúdo de schema.txt:

```

{
  "cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
    \"appliesTo\": {\"principalTypes\": [\"Employee\"]}}}, \"entityTypes\": {
    \"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\":
    \"Long\"}}, \"name\": {\"type\": \"String\"}}, \"type\": \"Record\"}}}}}"

```

```
}
```

Saída:

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "namespaces": [
    "MySampleNamespace"
  ],
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

Para obter mais informações sobre o esquema, consulte Esquema do [repositório de políticas no Guia](#) do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [PutSchema](#) na Referência de AWS CLI Comandos.

update-identity-source

O código de exemplo a seguir mostra como usar `update-identity-source`.

AWS CLI

Para atualizar uma fonte de identidade

O `update-identity-source` exemplo a seguir modifica a fonte de identidade especificada fornecendo uma nova configuração do grupo de usuários do Cognito e alterando o tipo de entidade retornado pela fonte de identidade.

```
aws verifiedpermissions update-identity-source
--identity-source-id ISEXAMPLEabcdefg111111 \
--update-configuration file://config.txt \
--principal-entity-type "Employee" \
--policy-store-id PSEXAMPLEabcdefg111111
```

Conteúdo de `config.txt`:

```
{
  "cognitoUserPoolConfiguration": {
```



```
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
    }
}
```

Saída:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEabcdefgh111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEabcdefgh111111"
}
```

Para obter mais informações sobre fontes de identidade, consulte [Uso de permissões verificadas da Amazon com provedores de identidade](#) no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [UpdateIdentitySource](#) na Referência de AWS CLI Comandos.

update-policy-store

O código de exemplo a seguir mostra como usar update-policy-store.

AWS CLI

Para atualizar um repositório de políticas

O update-policy-store exemplo a seguir modifica um repositório de políticas alterando sua configuração de validação.

```
aws verifiedpermissions update-policy-store \
  --validation-settings "mode=STRICT" \
  --policy-store-id PSEXAMPLEabcdefgh111111
```

Saída:

```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefgh111111",
```

```
"createdDate": "2023-05-16T17:41:29.103459+00:00",  
"lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",  
"policyStoreId": "PSEXAMPLEEabcdefg111111"  
}
```

Para obter mais informações sobre repositórios de políticas, consulte Armazenamentos de [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [UpdatePolicyStore](#) na Referência de AWS CLI Comandos.

update-policy-template

O código de exemplo a seguir mostra como usar update-policy-template.

AWS CLI

Exemplo 1: Para atualizar um modelo de política

O update-policy-template exemplo a seguir modifica a política vinculada ao modelo especificado para substituir sua declaração de política.

```
aws verifiedpermissions update-policy-template \  
  --policy-template-id PTEXAMPLEEabcdefg111111 \  
  --statement file://template1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Conteúdo do arquivo template1.txt:

```
permit(  
  principal in ?principal,  
  action == Action::"view",  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Saída:

```
{  
  "createdDate": "2023-06-12T20:47:42.804511+00:00",  
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
```

```
"policyTemplateId": "PTEXAMPLEabcdefg111111"  
}
```

Para obter mais informações sobre modelos de políticas, consulte Modelos de [política de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [UpdatePolicyTemplate](#) na Referência de AWS CLI Comandos.

update-policy

O código de exemplo a seguir mostra como usar `update-policy`.

AWS CLI

Exemplo 1: Para criar uma política estática

O `create-policy` exemplo a seguir cria uma política estática com um escopo de política que especifica tanto um principal quanto um recurso.

```
aws verifiedpermissions create-policy \  
  --definition file://definition.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

O `statement` parâmetro usa uma representação em cadeia de caracteres de um JSON objeto. Ele contém aspas incorporadas (") dentro do par de aspas mais externo. Isso exige que você converta o JSON em uma string precedendo todas as aspas incorporadas com um caractere de barra invertida (\) e combinando todas as linhas em uma única linha de texto sem quebras de linha.

As sequências de caracteres de exemplo podem ser exibidas em várias linhas aqui para facilitar a leitura, mas a operação exige que os parâmetros sejam enviados como cadeias de caracteres de uma única linha.

Conteúdo do arquivo `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\" );"
  }
}

```

Saída:

```

{
  "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"
  }
}

```

Exemplo 2: Para criar uma política estática que conceda acesso a um recurso para todos

O `create-policy` exemplo a seguir cria uma política estática com um escopo de política que especifica somente um recurso.

```

aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEabcdefg111111

```

Conteúdo do arquivo `definition2.txt`:

```

{
  "static": {
    "description": "Grant everyone access to the publicFolder Album",
    "statement": "permit(principal, action, resource in Album::\\"publicFolder
\");"
  }
}

```

Saída:

```
{
  "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"
  }
}
```

Exemplo 3: Para criar uma política vinculada ao modelo que esteja associada ao modelo especificado

O `create-policy` exemplo a seguir cria uma política vinculada ao modelo usando o modelo de política especificado e associa o principal especificado a ser usado à nova política vinculada ao modelo.

```
aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Conteúdo do `definition3.txt`:

```
{
  "templateLinked": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    }
  }
}
```

Saída:

```
{
```

```
"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"
}
}
```

Para obter mais informações sobre políticas, consulte as [políticas de permissões verificadas](#) da Amazon no Guia do usuário de permissões verificadas da Amazon.

- Para API obter detalhes, consulte [UpdatePolicy](#) na Referência de AWS CLI Comandos.

VPCExemplos de treliça usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with VPC Lattice.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-listener

O código de exemplo a seguir mostra como usar `create-listener`.

AWS CLI

Para criar um ouvinte

O `create-listener` exemplo a seguir cria um HTTPS ouvinte com uma regra padrão que encaminha o tráfego para o grupo-alvo do VPC Lattice especificado.

```
aws vpc-lattice create-listener \  
  --name my-service-listener \  
  --protocol HTTPS \  
  --port 443 \  
  --service-identifier svc-0285b53b2eEXAMPLE \  
  --default-action file://listener-config.json
```

Conteúdo de `listener-config.json`:

```
{  
  "forward": {  
    "targetGroups": [  
      {  
        "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"  
      }  
    ]  
  }  
}
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Listeners no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [CreateListener](#) na Referência de AWS CLI Comandos.

create-service-network-service-association

O código de exemplo a seguir mostra como usar `create-service-network-service-association`.

AWS CLI

Para criar uma associação de serviço

O `create-service-network-service-association` exemplo a seguir associa o serviço especificado à rede de serviços especificada.

```
aws vpc-lattice create-service-network-service-association \
  --service-identifier svc-0285b53b2eEXAMPLE \
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

Saída:

```
{
  "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"
```



```
}
```

Para obter mais informações, consulte [Gerenciar associações de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [CreateServiceNetworkServiceAssociation](#) na Referência de AWS CLI Comandos.

create-service-network-vpc-association

O código de exemplo a seguir mostra como usar `create-service-network-vpc-association`.

AWS CLI

Para criar uma VPC associação

O `create-service-network-vpc-association` exemplo a seguir associa a vpc especificada à rede de serviços especificada. O grupo de segurança especificado controla quais recursos do VPC podem acessar a rede de serviços e seus serviços.

```
aws vpc-lattice create-service-network-vpc-association \  
  --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --security-group-ids sg-0aee16bc6cEXAMPLE
```

Saída:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/  
snva-0821fc8631EXAMPLE",  
  "createdBy": "123456789012",  
  "id": "snva-0821fc8631EXAMPLE",  
  "securityGroupIds": [  
    "sg-0aee16bc6cEXAMPLE"  
  ],  
  "status": "CREATE_IN_PROGRESS"  
}
```

Para obter mais informações, consulte [Gerenciar VPC associações](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [CreateServiceNetworkVpcAssociation](#) na Referência de AWS CLI Comandos.

create-service-network

O código de exemplo a seguir mostra como usar `create-service-network`.

AWS CLI

Para criar uma rede de serviços

O `create-service-network` exemplo a seguir cria uma rede de serviços com o nome especificado.

```
aws vpc-lattice create-service-network \  
  --name my-service-network
```

Saída:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/  
sn-080ec7dc93EXAMPLE",  
  "authType": "NONE",  
  "id": "sn-080ec7dc93EXAMPLE",  
  "name": "my-service-network"  
}
```

Para obter mais informações, consulte [Redes de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [CreateServiceNetwork](#) na Referência de AWS CLI Comandos.

create-service

O código de exemplo a seguir mostra como usar `create-service`.

AWS CLI

Para criar um serviço

O `create-service` exemplo a seguir cria um serviço com o nome especificado.

```
aws vpc-lattice create-service \  
  --name my-lattice-service
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Services in VPC Lattice no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [CreateService](#) na Referência de AWS CLI Comandos.

`create-target-group`

O código de exemplo a seguir mostra como usar `create-target-group`.

AWS CLI

Exemplo 1: Para criar um grupo-alvo do tipo `INSTANCE`

O `create-target-group` exemplo a seguir cria um grupo-alvo com o nome, o tipo e a configuração especificados.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-instance \  
  --type INSTANCE \  
  --config file://tg-config.json
```

Conteúdo de tg-config.json:

```
{
  "port": 443,
  "protocol": "HTTPS",
  "protocolVersion": "HTTP1",
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

Saída:

```
{
  "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"
}
```

Exemplo 2: Para criar um grupo-alvo do tipo IP

O `create-target-group` exemplo a seguir cria um grupo-alvo com o nome, o tipo e a configuração especificados.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-ip \  
  --type IP \  
  --config file://tg-config.json
```

Conteúdo de tg-config.json:

```
{  
  "ipAddressType": "IPV4",  
  "port": 443,  
  "protocol": "HTTPS",  
  "protocolVersion": "HTTP1",  
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

Saída:

```
{  
  "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"
}
```

Exemplo 3: Para criar um grupo-alvo do tipo LAMBDA

O `create-target-group` exemplo a seguir cria um grupo-alvo com o nome, o tipo e a configuração especificados.

```
aws vpc-lattice create-target-group \
  --name my-lattice-target-group-lambda \
  --type LAMBDA
```

Saída:

```
{
  "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"
}
```

Exemplo 4: Para criar um grupo-alvo do tipo ALB

O `create-target-group` exemplo a seguir cria um grupo-alvo com o nome, o tipo e a configuração especificados.

```
aws vpc-lattice create-target-group \
  --name my-lattice-target-group-alb \
  --type ALB \
  --config file://tg-config.json
```

Conteúdo de `tg-config.json`:

```
{
  "port": 443,
  "protocol": "HTTPS",
  "protocolVersion": "HTTP1",
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
```

```
}
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [CreateTargetGroup](#) na Referência de AWS CLI Comandos.

delete-auth-policy

O código de exemplo a seguir mostra como usar delete-auth-policy.

AWS CLI

Para excluir uma política de autenticação

O delete-auth-policy exemplo a seguir exclui a política de autenticação para o serviço especificado.

```
aws vpc-lattice delete-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [as políticas de autenticação no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteAuthPolicy](#) na Referência de AWS CLI Comandos.

delete-listener

O código de exemplo a seguir mostra como usar delete-listener.

AWS CLI

Para excluir um ouvinte

O delete-listener exemplo a seguir exclui o ouvinte especificado.

```
aws vpc-lattice delete-listener \  
  --listener-identifier listener-07cc7fb0abEXAMPLE \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Listeners no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteListener](#) na Referência de AWS CLI Comandos.

delete-service-network-service-association

O código de exemplo a seguir mostra como usar delete-service-network-service-association.

AWS CLI

Para excluir uma associação de serviço

O delete-service-network-service-association exemplo a seguir desassocia a associação de serviço especificada.

```
aws vpc-lattice delete-service-network-service-association \  
  --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

Saída:

```
{  
  "arn": "arn:aws:vpc-lattice:us-  
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",  
  "id": "snsa-031fabb4d8EXAMPLE",
```



```
"status": "DELETE_IN_PROGRESS"
}
```

Para obter mais informações, consulte [Gerenciar associações de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteServiceNetworkServiceAssociation](#) na Referência de AWS CLI Comandos.

delete-service-network-vpc-association

O código de exemplo a seguir mostra como usar `delete-service-network-vpc-association`.

AWS CLI

Para excluir uma VPC associação

O `delete-service-network-vpc-association` exemplo a seguir dissocia a associação especificada VPC.

```
aws vpc-lattice delete-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

Saída:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
  "id": "snva-0821fc8631EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

Para obter mais informações, consulte [Gerenciar VPC associações](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteServiceNetworkVpcAssociation](#) na Referência de AWS CLI Comandos.

delete-service-network

O código de exemplo a seguir mostra como usar `delete-service-network`.

AWS CLI

Para excluir uma rede de serviços

O `delete-service-network` exemplo a seguir exclui a rede de serviços especificada.

```
aws vpc-lattice delete-service-network \  
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

Este comando não produz saída.

Para obter mais informações, consulte [Redes de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteServiceNetwork](#) na Referência de AWS CLI Comandos.

`delete-service`

O código de exemplo a seguir mostra como usar `delete-service`.

AWS CLI

Como excluir um serviço

O `delete-service` exemplo a seguir exclui o serviço especificado.

```
aws vpc-lattice delete-service \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Saída:

```
{  
  "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/  
svc-0285b53b2eEXAMPLE",  
  "id": "svc-0285b53b2eEXAMPLE",  
  "name": "my-lattice-service",  
  "status": "DELETE_IN_PROGRESS"  
}
```

Para obter mais informações, consulte [Services in VPC Lattice no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteService](#) na Referência de AWS CLI Comandos.

delete-target-group

O código de exemplo a seguir mostra como usar `delete-target-group`.

AWS CLI

Como excluir um grupo de destino

O exemplo de `delete-target-group` a seguir exclui o grupo de destino especificado.

```
aws vpc-lattice delete-target-group \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Saída:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "status": "DELETE_IN_PROGRESS"  
}
```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeleteTargetGroup](#) na Referência de AWS CLI Comandos.

deregister-targets

O código de exemplo a seguir mostra como usar `deregister-targets`.

AWS CLI

Para cancelar o registro de um alvo

O `deregister-targets` exemplo a seguir cancela o registro do alvo especificado do grupo de destino especificado.

```
aws vpc-lattice deregister-targets \  
  --targets i-07dd579bc5EXAMPLE \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Saída:

```
{
  "successful": [
    {
      "id": "i-07dd579bc5EXAMPLE",
      "port": 443
    }
  ],
  "unsuccessful": []
}
```

Para obter mais informações, consulte [Registrar alvos](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [DeregisterTargets](#) na Referência de AWS CLI Comandos.

get-auth-policy

O código de exemplo a seguir mostra como usar `get-auth-policy`.

AWS CLI

Para obter informações sobre uma política de autenticação

O `get-auth-policy` exemplo a seguir obtém informações sobre a política de autenticação para o serviço especificado.

```
aws vpc-lattice get-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [as políticas de autenticação no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetAuthPolicy](#) na Referência de AWS CLI Comandos.

get-listener

O código de exemplo a seguir mostra como usar `get-listener`.

AWS CLI

Para obter informações sobre um ouvinte de serviço

O `get-listener` exemplo a seguir obtém informações sobre o ouvinte especificado para o serviço especificado.

```
aws vpc-lattice get-listener \  
  --listener-identifier listener-0ccf55918cEXAMPLE \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Definir roteamento no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetListener](#) na Referência de AWS CLI Comandos.

get-service-network-service-association

O código de exemplo a seguir mostra como usar `get-service-network-service-association`.

AWS CLI

Para obter informações sobre uma associação de serviços

O `get-service-network-service-association` exemplo a seguir obtém informações sobre a associação de serviço especificada.

```
aws vpc-lattice get-service-network-service-association \  
  --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

Saída:

```
{  
  "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"
}
```

Para obter mais informações, consulte [Gerenciar associações de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetServiceNetworkServiceAssociation](#) na Referência de AWS CLI Comandos.

get-service-network-vpc-association

O código de exemplo a seguir mostra como usar `get-service-network-vpc-association`.

AWS CLI

Para obter informações sobre uma VPC associação

O `get-service-network-vpc-association` exemplo a seguir obtém informações sobre a VPC associação especificada.

```
aws vpc-lattice get-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte [Gerenciar VPC associações](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetServiceNetworkVpcAssociation](#) na Referência de AWS CLI Comandos.

get-service-network

O código de exemplo a seguir mostra como usar `get-service-network`.

AWS CLI

Para obter informações sobre uma rede de serviços

O `get-service-network` exemplo a seguir obtém informações sobre a rede de serviços especificada.

```
aws vpc-lattice get-service-network \
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

Saída:

```
{
  "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
}
```

Para obter mais informações, consulte [Redes de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetServiceNetwork](#) na Referência de AWS CLI Comandos.

get-service

O código de exemplo a seguir mostra como usar `get-service`.

AWS CLI

Para obter informações sobre um serviço

O `get-service` exemplo a seguir obtém informações sobre o serviço especificado.

```
aws vpc-lattice get-service \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetService](#) na Referência de AWS CLI Comandos.

get-target-group

O código de exemplo a seguir mostra como usar `get-target-group`.

AWS CLI

Para obter informações sobre um grupo-alvo

O `get-target-group` exemplo a seguir obtém informações sobre o grupo-alvo especificado, que tem um tipo de alvo de `INSTANCE`.

```
aws vpc-lattice get-target-group \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [GetTargetGroup](#) na Referência de AWS CLI Comandos.

list-listeners

O código de exemplo a seguir mostra como usar `list-listeners`.

AWS CLI

Para listar os ouvintes do serviço

O `list-listeners` exemplo a seguir lista os ouvintes do serviço especificado.

```
aws vpc-lattice list-listeners \
  --service-identifier svc-0285b53b2eEXAMPLE
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Definir roteamento no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListListeners](#) na Referência de AWS CLI Comandos.

list-service-network-service-associations

O código de exemplo a seguir mostra como usar `list-service-network-service-associations`.

AWS CLI

Para listar associações de serviços

O `list-service-network-service-associations` exemplo a seguir lista as associações de serviço para a rede de serviços especificada. A `--query` opção define o escopo da saída para as associações IDs de serviços.

```
aws vpc-lattice list-service-network-service-associations \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --query items[*].id
```

Saída:

```
[  
  "snsa-031fabb4d8EXAMPLE",  
  "snsa-0e16955a8cEXAMPLE"  
]
```

Para obter mais informações, consulte [Gerenciar associações de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListServiceNetworkServiceAssociations](#) na Referência de AWS CLI Comandos.

list-service-network-vpc-associations

O código de exemplo a seguir mostra como usar `list-service-network-vpc-associations`.

AWS CLI

Para listar VPC associações

O `list-service-network-vpc-associations` exemplo a seguir lista as VPC associações para a rede de serviços especificada. A `--query` opção define o escopo da saída para a IDs das VPC associações.

```
aws vpc-lattice list-service-network-vpc-associations \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --query items[*].id
```

Saída:

```
[
  "snva-0821fc8631EXAMPLE",
  "snva-0c5dcb60d6EXAMPLE"
]
```

Para obter mais informações, consulte [Gerenciar VPC associações](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListServiceNetworkVpcAssociations](#) na Referência de AWS CLI Comandos.

list-service-networks

O código de exemplo a seguir mostra como usar `list-service-networks`.

AWS CLI

Para listar suas redes de serviços

O `list-service-networks` exemplo a seguir lista as redes de serviço pertencentes ou compartilhadas com a conta de chamada. A `--query` opção atribui os resultados aos Amazon Resource Names (ARN) das redes de serviços.

```
aws vpc-lattice list-service-networks \
  --query items[*].arn
```

Saída:

```
[
  "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
  "arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"
]
```

Para obter mais informações, consulte [Redes de serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListServiceNetworks](#) na Referência de AWS CLI Comandos.

list-services

O código de exemplo a seguir mostra como usar `list-services`.

AWS CLI

Para listar seus serviços

O `list-services` exemplo a seguir lista os serviços de propriedade ou compartilhados com a conta de chamada. A `--query` opção atribui os resultados aos Amazon Resource Names (ARN) dos serviços.

```
aws vpc-lattice list-services \  
  --query items[*].arn
```

Saída:

```
[  
  "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",  
  "arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"  
]
```

Para obter mais informações, consulte [Serviços](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListServices](#) na Referência de AWS CLI Comandos.

list-target-groups

O código de exemplo a seguir mostra como usar `list-target-groups`.

AWS CLI

Para listar seus grupos-alvo

O `list-target-groups` exemplo a seguir lista os grupos-alvo com um tipo de alvo de LAMBDA.

```
aws vpc-lattice list-target-groups \  
  --target-group-type LAMBDA
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListTargetGroups](#) na Referência de AWS CLI Comandos.

list-targets

O código de exemplo a seguir mostra como usar `list-targets`.

AWS CLI

Para listar os alvos de um grupo-alvo

O `list-targets` exemplo a seguir lista os alvos do grupo-alvo especificado.

```
aws vpc-lattice list-targets \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Saída:

```
{
  "items": [
    {
      "id": "i-07dd579bc5EXAMPLE",
```

```

        "port": 443,
        "status": "HEALTHY"
    },
    {
        "id": "i-047b3c9078EXAMPLE",
        "port": 443,
        "reasonCode": "HealthCheckFailed",
        "status": "UNHEALTHY"
    }
]
}

```

Para obter mais informações, consulte [Grupos-alvo](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [ListTargets](#) na Referência de AWS CLI Comandos.

put-auth-policy

O código de exemplo a seguir mostra como usar `put-auth-policy`.

AWS CLI

Para criar uma política de autenticação para um serviço

O `put-auth-policy` exemplo a seguir concede acesso às solicitações de qualquer principal autenticado que use a IAM função especificada. O recurso é o ARN serviço ao qual a política está vinculada.

```

aws vpc-lattice put-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE \
  --policy file://auth-policy.json

```

Conteúdo de `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"
    }
]
}

```

Saída:

```

{
  "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"
}

```

Para obter mais informações, consulte [as políticas de autenticação no Guia](#) do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [PutAuthPolicy](#) na Referência de AWS CLI Comandos.

register-targets

O código de exemplo a seguir mostra como usar `register-targets`.

AWS CLI

Para registrar um alvo

O `register-targets` exemplo a seguir registra os alvos especificados com o grupo de destino especificado.

```

aws vpc-lattice register-targets \
  --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE

```

Saída:

```

{
  "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
    }
  ]
}
```

Para obter mais informações, consulte [Registrar alvos](#) no Guia do usuário do Amazon VPC Lattice.

- Para API obter detalhes, consulte [RegisterTargets](#) na Referência de AWS CLI Comandos.

AWS WAF Classic exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS WAF Classic.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

put-logging-configuration

O código de exemplo a seguir mostra como usar `put-logging-configuration`.

AWS CLI

Para criar uma configuração de registro para a web ACL ARN com o stream especificado do Kinesis Firehose ARN

O `put-logging-configuration` exemplo a seguir exibe a configuração de registro para WAF with CloudFront.

```
aws waf put-logging-configuration \
  --logging-configuration ResourceArn=arn:aws:waf::123456789012:webacl/3bffd3ed-
fa2e-445e-869f-a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-
east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[]
```

Saída:

```
{
  "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"
    ]
  }
}
```

- Para API obter detalhes, consulte [PutLoggingConfiguration](#) na Referência de AWS CLI Comandos.

update-byte-match-set

O código de exemplo a seguir mostra como usar `update-byte-match-set`.

AWS CLI

Para atualizar um conjunto de correspondências de bytes

O `update-byte-match-set` comando a seguir exclui um `ByteMatchTuple` objeto (filtro) em um `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
```

Para obter mais informações, consulte [Trabalhando com condições de correspondência de strings](#) no guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateByteMatchSet](#) na Referência de AWS CLI Comandos.

update-ip-set

O código de exemplo a seguir mostra como usar `update-ip-set`.

AWS CLI

Para atualizar um conjunto de IP

O `update-ip-set` comando a seguir atualiza um IPSet com um IPv4 endereço e exclui um IPv6 endereço:

```
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
```

Como alternativa, você pode usar um JSON arquivo para especificar a entrada. Por exemplo:

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-
token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates file://change.json
```

Onde está o conteúdo JSON do arquivo:

```
[
{
  "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"
}
}
]
```

Para obter mais informações, consulte [Trabalhando com condições de correspondência de IP](#) no guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateIpSet](#) na Referência de AWS CLI Comandos.

update-rule

O código de exemplo a seguir mostra como usar `update-rule`.

AWS CLI

Para atualizar uma regra

O `update-rule` comando a seguir exclui um objeto Predicate em uma regra:

```
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"}
```

Para obter mais informações, consulte [Como trabalhar com regras](#) no guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateRule](#) na Referência de AWS CLI Comandos.

update-size-constraint-set

O código de exemplo a seguir mostra como usar `update-size-constraint-set`.

AWS CLI

Para atualizar um conjunto de restrições de tamanho

O `update-size-constraint-set` comando a seguir exclui um SizeConstraint objeto (filtros) em um conjunto de restrições de tamanho:

```
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"
```

Para obter mais informações, consulte [Trabalhando com condições de restrição de tamanho](#) no guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateSizeConstraintSet](#)na Referência de AWS CLI Comandos.

update-sql-injection-match-set

O código de exemplo a seguir mostra como usar `update-sql-injection-match-set`.

AWS CLI

Para atualizar um conjunto de SQL fósforos de injeção

O `update-sql-injection-match-set` comando a seguir exclui um `SqlInjectionMatchTuple` objeto (filtros) em um conjunto de SQL correspondências de injeção:

```
aws waf update-sql-injection-match-set --sql-injection-  
match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --  
change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates  
Action="DELETE",SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="NONE"
```

Para obter mais informações, consulte [Trabalhando com condições de correspondência de SQL injeção](#) no guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateSqlInjectionMatchSet](#)na Referência de AWS CLI Comandos.

update-web-acl

O código de exemplo a seguir mostra como usar `update-web-acl`.

AWS CLI

Para atualizar uma web ACL

O `update-web-acl` comando a seguir exclui um `ActivatedRule` objeto em uma `WebACL`.

```
aws waf update-web-acl -- web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token
12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates Action=" ", =' {Priority=1, =' -1-Exemplo",
Action= {Type=" "}, Type=" "} 'DELETEActivatedRuleRuleIdWAFRuleALLOWREGULAR
```

Saída:

```
{
  "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
}
```

Para obter mais informações, consulte [Working with Web ACLs](#) in the AWS WAF, AWS Firewall Manager e AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateWebAcl](#) na Referência de AWS CLI Comandos.

update-xss-match-set

O código de exemplo a seguir mostra como usar `update-xss-match-set`.

AWS CLI

Para atualizar um `XSSMatchSet`

O `update-xss-match-set` comando a seguir exclui um `XssMatchTuple` objeto (filtros) em um `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
```

Para obter mais informações, consulte [Como trabalhar com condições de correspondência de scripts entre sites](#) no guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateXssMatchSet](#) na Referência de AWS CLI Comandos.

AWS WAF Classic regional exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS WAF Classic regional.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-web-acl

O código de exemplo a seguir mostra como usar `associate-web-acl`.

AWS CLI

Para associar uma web ACL a um recurso

O `associate-web-acl` comando a seguir associa uma webACL, especificada pelo, a um recurso `web-acl-id`, especificado pelo `resource-arn`. O recurso ARN pode se referir a um balanceador de carga de aplicativo ou a um API gateway:

```
aws waf-regional associate-web-acl \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

Para obter mais informações, consulte [Trabalhando com a Web ACLs](#) no Guia do AWS WAF Desenvolvedor.

- Para API obter detalhes, consulte [AssociateWebAcl](#) na Referência de AWS CLI Comandos.

put-logging-configuration

O código de exemplo a seguir mostra como usar `put-logging-configuration`.

AWS CLI

Para criar uma configuração de registro para a web ACL ARN com o stream especificado do Kinesis Firehose ARN

O `put-logging-configuration` exemplo a seguir exibe a configuração de registro para WAF with ALB/API Gateway in Region `us-east-1`.

```
aws waf-regional put-logging-configuration \
  --logging-configuration ResourceArn=arn:aws:waf-
regional:us-east-1:123456789012:webacl/3bffd3ed-fa2e-445e-869f-
a6a7cf153fd3,LogDestinationConfigs=arn:aws:firehose:us-
east-1:123456789012:deliverystream/aws-waf-logs-firehose-stream,RedactedFields=[] \
  --region us-east-1
```

Saída:

```
{
  "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"
    ]
  }
}
```

- Para API obter detalhes, consulte [PutLoggingConfiguration](#) na Referência de AWS CLI Comandos.

update-byte-match-set

O código de exemplo a seguir mostra como usar `update-byte-match-set`.

AWS CLI

Para atualizar um conjunto de correspondências de bytes

O `update-byte-match-set` comando a seguir exclui um `ByteMatchTuple` objeto (filtro) em um `ByteMatchSet`. Como o `updates` valor tem aspas duplas incorporadas, você deve colocar o valor entre aspas simples.

```
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="
```

Para obter mais informações, consulte Como [trabalhar com condições de correspondência de strings](#) no Guia do AWS WAF desenvolvedor.

- Para API obter detalhes, consulte [UpdateByteMatchSet](#) na Referência de AWS CLI Comandos.

update-ip-set

O código de exemplo a seguir mostra como usar `update-ip-set`.

AWS CLI

Para atualizar um conjunto de IPs

O `update-ip-set` comando a seguir atualiza um IPSet com um IPv4 endereço e exclui um IPv6 endereço. Obtenha o valor de `change-token` executando o `get-change-token` comando. Como o valor das atualizações inclui aspas duplas incorporadas, você deve colocar o valor entre aspas simples.

```
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
```

Como alternativa, você pode usar um JSON arquivo para especificar a entrada. Por exemplo:

```
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
```

Conteúdo do `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"
      }
    }
  ]

```

Para obter mais informações, consulte [Trabalhando com condições de correspondência de IP](#) no Guia do AWS WAF desenvolvedor.

- Para API obter detalhes, consulte [UpdateIpSet](#) na Referência de AWS CLI Comandos.

update-rule

O código de exemplo a seguir mostra como usar `update-rule`.

AWS CLI

Para atualizar uma regra

O `update-rule` comando a seguir exclui um Predicate objeto em uma regra. Como o `updates` valor tem aspas duplas incorporadas, você deve colocar o valor inteiro entre aspas simples.

```

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"}'

```

Para obter mais informações, consulte Como [trabalhar com regras](#) no Guia do AWS WAF desenvolvedor.

- Para API obter detalhes, consulte [UpdateRule](#) na Referência de AWS CLI Comandos.

update-size-constraint-set

O código de exemplo a seguir mostra como usar `update-size-constraint-set`.

AWS CLI

Para atualizar um conjunto de restrições de tamanho

O `update-size-constraint-set` comando a seguir exclui um objeto `SizeConstraint` (filtros) em um conjunto de restrições de tamanho. Como o `updates` valor contém aspas duplas incorporadas, você deve colocar o valor inteiro entre aspas simples.

```
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="NON"
```

Para obter mais informações, consulte [Trabalhando com condições de restrição de tamanho](#) no Guia do AWS WAF desenvolvedor.

- Para API obter detalhes, consulte [UpdateSizeConstraintSet](#) na Referência de AWS CLI Comandos.

update-sql-injection-match-set

O código de exemplo a seguir mostra como usar `update-sql-injection-match-set`.

AWS CLI

Para atualizar um conjunto de SQL fósforos de injeção

O `update-sql-injection-match-set` comando a seguir exclui um `SqlInjectionMatchTuple` objeto (filtros) em um conjunto de SQL correspondências de injeção. Como o `updates` valor contém aspas duplas incorporadas, você deve colocar o valor inteiro entre aspas simples. :

```
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 'Ação = " " , = { = {Type=" _ "}, = " _ "  
'DELETESqlInjectionMatchTupleFieldToMatchQUERYSTRINGTextTransformationURLDECODE
```

Para obter mais informações, consulte [Trabalhando com condições de correspondência de SQL injeção](#) no Guia do AWS WAF desenvolvedor.

- Para API obter detalhes, consulte [UpdateSqlInjectionMatchSet](#) na Referência de AWS CLI Comandos.

update-web-acl

O código de exemplo a seguir mostra como usar `update-web-acl`.

AWS CLI

Para atualizar uma web ACL

O `update-web-acl` comando a seguir exclui um `ActivatedRule` objeto em uma `WebACL`. Como o `updates` valor contém aspas duplas incorporadas, você deve colocar o valor inteiro entre aspas simples.

```
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"}'
```

Para obter mais informações, consulte [Trabalhando com a Web ACLs](#) no Guia do AWS WAF Desenvolvedor.

- Para API obter detalhes, consulte [UpdateWebAcl](#) na Referência de AWS CLI Comandos.

update-xss-match-set

O código de exemplo a seguir mostra como usar `update-xss-match-set`.

AWS CLI

Para atualizar um `XSSMatchSet`

O `update-xss-match-set` comando a seguir exclui um `XssMatchTuple` objeto (filtros) em um `XssMatchSet`. Como o `updates` valor contém aspas duplas incorporadas, você deve colocar o valor inteiro entre aspas simples.

```
aws waf-regional update-xss-match-set \  
  --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \  
  --updates XssMatchTuple='{Priority=1,RuleId="WAFRule-1-  
Example",Action={Type="ALLOW"},Type="ALLOW"}'
```

```
--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'
```

Para obter mais informações, consulte Como [trabalhar com condições de correspondência de scripts entre sites](#) no Guia do AWS WAFdesenvolvedor.

- Para API obter detalhes, consulte [UpdateXssMatchSet](#)na Referência de AWS CLI Comandos.

AWS WAFV2 exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with AWS WAFV2.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-web-acl

O código de exemplo a seguir mostra como usar `associate-web-acl`.

AWS CLI

Para associar uma web ACL a um AWS recurso regional

O `associate-web-acl` exemplo a seguir associa a web ACL especificada a um 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
```

Este comando não produz saída.

Para obter mais informações, consulte [Associando ou desassociando uma Web ACL a um AWS recurso](#) no AWS Firewall AWS WAF Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [AssociateWebAcl](#) na Referência de AWS CLI Comandos.

check-capacity

O código de exemplo a seguir mostra como usar `check-capacity`.

AWS CLI

Para obter a capacidade usada por um conjunto de regras

A seguir, são `check-capacity` recuperados os requisitos de capacidade de um conjunto de regras que contém uma declaração de regra baseada em taxa e uma declaração de AND regra que contém regras aninhadas.

```
aws wafv2 check-capacity \  
  --scope REGIONAL \  
  --rules file://waf-rule-list.json \  
  --region us-west-2
```

Conteúdo do arquivo://waf-rule-list.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"
}
},
{
  "Name":"rate-rule",
  "Priority":1,
  "Statement":{
    "RateBasedStatement":{
      "Limit":1000,
      "AggregateKeyType":"IP"
    }
  },
  "Action":{
    "Block":{
```



```

    }
  },
  "VisibilityConfig":{
    "SampledRequestsEnabled":true,
    "CloudWatchMetricsEnabled":true,
    "MetricName":"rate-rule"
  }
}
]

```

Saída:

```

{
  "Capacity":15
}

```

Para obter mais informações, consulte [AWS WAFWeb ACL Capacity Units \(WCU\)](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [CheckCapacity](#) na Referência de AWS CLI Comandos.

create-ip-set

O código de exemplo a seguir mostra como usar `create-ip-set`.

AWS CLI

Para criar um conjunto de IP para uso em sua web ACLs e grupos de regras

O `create-ip-set` comando a seguir cria um conjunto de IP com uma especificação de intervalo de endereços único.

```

aws wafv2 create-ip-set \
  --name testip \
  --scope REGIONAL \
  --ip-address-version IPV4 \
  --addresses 198.51.100.0/16

```

Saída:

```

{

```

```

    "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"
    }
  }
}

```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [CreateIpSet](#) na Referência de AWS CLI Comandos.

create-regex-pattern-set

O código de exemplo a seguir mostra como usar create-regex-pattern-set.

AWS CLI

Para criar um conjunto de padrões regex para uso em sua web ACLs e grupos de regras

O create-regex-pattern-set comando a seguir cria um conjunto de padrões regex com dois padrões de regex especificados.

```

aws wafv2 create-regex-pattern-set \
  --name regexPatterSet01 \
  --scope REGIONAL \
  --description 'Test web-acl' \
  --regular-expression-list '[{"RegexString": "/[0-9]*/"}, {"RegexString": "/[a-z]*/"}]'

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [CreateRegexPatternSet](#) na Referência de AWS CLI Comandos.

create-rule-group

O código de exemplo a seguir mostra como usar `create-rule-group`.

AWS CLI

Para criar um grupo de regras personalizado para uso em sua web ACLs

O `create-rule-group` comando a seguir cria um grupo de regras personalizado para uso regional. As instruções de regra para o grupo são fornecidas em um arquivo JSON formatado.

```

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

```

Conteúdo do arquivo: `//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"
}
}
]

```

Saída:

```

{
  "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"
  }
}

```

Para obter mais informações, consulte [Gerenciando seus próprios grupos de regras](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [CreateRuleGroup](#) na Referência de AWS CLI Comandos.

create-web-acl

O código de exemplo a seguir mostra como usar `create-web-acl`.

AWS CLI

Para criar uma web ACL

O `create-web-acl` comando a seguir cria uma web ACL para uso regional. As declarações de regras para a web ACL são fornecidas em um arquivo JSON formatado.

```

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

```

Conteúdo do arquivo: `//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"
    }
  }
]
```

Saída:

```
{
  "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"
  }
}
```

Para obter mais informações, consulte [Gerenciando e usando uma lista de controle de acesso à Web \(WebACL\)](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [CreateWebAcl](#) na Referência de AWS CLI Comandos.

delete-ip-set

O código de exemplo a seguir mostra como usar delete-ip-set.

AWS CLI

Para excluir um conjunto de IP

O seguinte delete-ip-set exclui o conjunto de IP especificado. Essa chamada requer um ID, que você pode obter da chamada list-ip-sets, e um token de bloqueio, que você pode obter das chamadas list-ip-sets get-ip-set e.

```
aws wafv2 delete-ip-set \
  --name test1 \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --lock-token 46851772-db6f-459d-9385-49428812e357
```

Este comando não produz saída.

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeleteIpSet](#) na Referência de AWS CLI Comandos.

delete-logging-configuration

O código de exemplo a seguir mostra como usar delete-logging-configuration.

AWS CLI

Para desativar o registro em uma web ACL

O seguinte delete-logging-configuration remove qualquer configuração de registro da web especificadaACL.

```
aws wafv2 delete-logging-configuration \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Este comando não produz saída.

Para obter mais informações, consulte [Registro de informações de ACL tráfego da Web](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeleteLoggingConfiguration](#) na Referência de AWS CLI Comandos.

delete-regex-pattern-set

O código de exemplo a seguir mostra como usar delete-regex-pattern-set.

AWS CLI

Para excluir um conjunto de padrões regex

O seguinte delete-regex-pattern-set atualiza as configurações do conjunto de padrões regex especificado. Essa chamada requer um ID, que você pode obter da chamada list-regex-pattern-sets, e um token de bloqueio, que você pode obter da chamada list-regex-pattern-sets ou da chamada 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
```


Este comando não produz saída.

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeleteRegexPatternSet](#) na Referência de AWS CLI Comandos.

delete-rule-group

O código de exemplo a seguir mostra como usar `delete-rule-group`.

AWS CLI

Para excluir um grupo de regras personalizado

O seguinte `delete-rule-group` exclui o grupo de regras personalizadas especificado. Essa chamada requer um ID, que você pode obter da chamada `list-rule-groups`, e um token de bloqueio, que você pode obter da chamada `list-rule-groups` ou da chamada `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciando seus próprios grupos de regras](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeleteRuleGroup](#) na Referência de AWS CLI Comandos.

delete-web-acl

O código de exemplo a seguir mostra como usar `delete-web-acl`.

AWS CLI

Para excluir uma web ACL

O seguinte `delete-web-acl` exclui a web especificada ACL da sua conta. Uma web só ACL pode ser excluída quando não está associada a nenhum recurso. Essa chamada requer um ID, que você pode obter da chamada `list-web-acls`, e um token de bloqueio, que você pode obter da chamada `list-web-acls` ou da chamada `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
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciando e usando uma lista de controle de acesso à Web \(WebACL\)](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DeleteWebAcl](#) na Referência de AWS CLI Comandos.

describe-managed-rule-group

O código de exemplo a seguir mostra como usar `describe-managed-rule-group`.

AWS CLI

Para recuperar a descrição de um grupo de regras gerenciadas

O seguinte `describe-managed-rule-group` recupera a descrição de um grupo de regras AWS gerenciadas.

```
aws wafv2 describe-managed-rule-group \  
  --vendor-name AWS \  
  --name AWSManagedRulesCommonRuleSet \  
  --scope REGIONAL
```

Saída:

```
{  
  "Capacity": 700,  
  "Rules": [  
    {  
      "Name": "NoUserAgent_HEADER",
```

```
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "UserAgent_BadBots_HEADER",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_QUERYSTRING",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_Cookie_HEADER",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "SizeRestrictions_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_COOKIE",
    "Action": {
      "Block": {}
    }
  }
}
```

```
    }
  },
  {
    "Name": "EC2MetaDataSSRF_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2MetaDataSSRF_QUERY_ARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_QUERY_ARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "RestrictedExtensions_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "RestrictedExtensions_QUERY_ARGUMENTS",
    "Action": {
      "Block": {}
    }
  }
}
```

```
    },
    {
      "Name": "GenericRFI_QUERYARGUMENTS",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "GenericRFI_BODY",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "GenericRFI_URI_PATH",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "CrossSiteScripting_COOKIE",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "CrossSiteScripting_QUERYARGUMENTS",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "CrossSiteScripting_BODY",
      "Action": {
        "Block": {}
      }
    },
    {
      "Name": "CrossSiteScripting_URI_PATH",
      "Action": {
        "Block": {}
      }
    }
  ]
```

```
}
```

Para obter mais informações, consulte [Grupos de regras gerenciados](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DescribeManagedRuleGroup](#) na Referência de AWS CLI Comandos.

disassociate-web-acl

O código de exemplo a seguir mostra como usar `disassociate-web-acl`.

AWS CLI

Para desassociar uma web ACL de um recurso regional AWS

O `disassociate-web-acl` exemplo a seguir remove qualquer ACL associação web existente do Application Load Balancer especificado.

```
aws wafv2 disassociate-web-acl \  
  --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/waf-cli-alb/1ea17125f8b25a2a \  
  --region us-west-2
```

Este comando não produz saída.

Para obter mais informações, consulte [Associando ou desassociando uma Web ACL a um AWS recurso](#) no AWS Firewall AWS WAF Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [DisassociateWebAcl](#) na Referência de AWS CLI Comandos.

get-ip-set

O código de exemplo a seguir mostra como usar `get-ip-set`.

AWS CLI

Para recuperar um conjunto de IP específico

O código a seguir `get-ip-set` recupera o conjunto de IP com o nome, escopo e ID especificados. Você pode obter o ID de um conjunto de IP a partir dos comandos `create-ip-set` `list-ip-sets` e.

```
aws wafv2 get-ip-set \  
  --name testip \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{  
  "IPSet":{  
    "Description": "",  
    "Name": "testip",  
    "IPAddressVersion": "IPV4",  
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Addresses": [  
      "192.0.2.0/16"  
    ]  
  },  
  "LockToken": "447e55ac-2396-4c6d-b9f9-86b67c17f8b5"  
}
```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetIpSet](#) na Referência de AWS CLI Comandos.

get-logging-configuration

O código de exemplo a seguir mostra como usar `get-logging-configuration`.

AWS CLI

Para recuperar as configurações de registro para uma web ACL

O seguinte `get-logging-configuration` recupera a configuração de registro para a web ACL especificada.

```
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
```

Saída:

```
{
  "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"
    ]
  }
}
```

Para obter mais informações, consulte [Registro de informações de ACL tráfego da Web](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetLoggingConfiguration](#) na Referência de AWS CLI Comandos.

get-rate-based-statement-managed-keys

O código de exemplo a seguir mostra como usar `get-rate-based-statement-managed-keys`.

AWS CLI

Para recuperar uma lista de endereços IP bloqueados por uma regra baseada em taxas

O seguinte `get-rate-based-statement-managed-keys` recupera os endereços IP atualmente bloqueados por uma regra baseada em taxas que está sendo usada para um aplicativo regional.

```
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
```

Saída:

```
{
  "ManagedKeysIPV4":{
    "IPAddressVersion":"IPV4",
    "Addresses":[
      "198.51.100.0/32"
    ]
  },
  "ManagedKeysIPV6":{
    "IPAddressVersion":"IPV6",
    "Addresses":[
    ]
  }
}
```

Para obter mais informações, consulte [Declaração de regra baseada em taxas](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetRateBasedStatementManagedKeys](#) na Referência de AWS CLI Comandos.

get-regex-pattern-set

O código de exemplo a seguir mostra como usar `get-regex-pattern-set`.

AWS CLI

Para recuperar um conjunto específico de padrões de regex

O seguinte `get-regex-pattern-set` recupera o padrão de regex definido com o nome, escopo, região e ID especificados. Você pode obter o ID de um padrão regex definido a partir dos comandos `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
```

Saída:

```
{
  "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"
}
```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetRegexPatternSet](#) na Referência de AWS CLI Comandos.

get-rule-group

O código de exemplo a seguir mostra como usar `get-rule-group`.

AWS CLI

Para recuperar um grupo específico de regras personalizadas

O seguinte `get-rule-group` recupera o grupo de regras personalizadas com o nome, escopo e ID especificados. Você pode obter o ID de um grupo de regras a partir dos comandos `create-rule-group` `list-rule-groups` e.

```
aws wafv2 get-rule-group \
  --name ff \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte [Gerenciando seus próprios grupos de regras](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetRuleGroup](#) na Referência de AWS CLI Comandos.

get-sampled-requests

O código de exemplo a seguir mostra como usar `get-sampled-requests`.

AWS CLI

Para recuperar uma amostra de solicitações da web para uma web ACL

A seguir, são `get-sampled-requests` recuperados os exemplos de solicitações da web para a webACL, a métrica da regra e o período de tempo especificados.

```

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

```

Saída:

```

{
  "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
}

```

Para obter mais informações, consulte [Exibindo uma amostra de solicitações da Web](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetSampledRequests](#) na Referência de AWS CLI Comandos.

get-web-acl-for-resource

O código de exemplo a seguir mostra como usar `get-web-acl-for-resource`.

AWS CLI

Para recuperar a web associada ACL a um recurso AWS

O seguinte `get-web-acl-for-resource` recupera o JSON para a web ACL que está associado ao recurso especificado.

```

aws wafv2 get-web-acl-for-resource \
  --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/waf-cli-alb/1ea17125f8b25a2a

```

Saída:

```

{
  "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"
}
}

```

Para obter mais informações, consulte [Associando ou desassociando uma Web ACL a um AWS recurso](#) no AWS Firewall AWS WAF Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetWebAclForResource](#) na Referência de AWS CLI Comandos.

get-web-acl

O código de exemplo a seguir mostra como usar `get-web-acl`.

AWS CLI

Para recuperar uma web ACL

O seguinte `get-web-acl` recupera a web ACL com o nome, escopo e ID especificados. Você pode obter o ID de uma web a ACL partir dos comandos `create-web-acl` `list-web-acls` e.

```

aws wafv2 get-web-acl \
  --name test01 \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```
{
  "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":"dGVzdHN0cmlyZw==",
                  "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"
}

```

Para obter mais informações, consulte [Gerenciando e usando uma lista de controle de acesso à Web \(WebACL\)](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [GetWebAcl](#) na Referência de AWS CLI Comandos.

list-available-managed-rule-groups

O código de exemplo a seguir mostra como usar `list-available-managed-rule-groups`.

AWS CLI

Para recuperar os grupos de regras gerenciados

O texto a seguir `list-available-managed-rule-groups` retorna a lista de todos os grupos de regras gerenciados que estão atualmente disponíveis para uso em sua webACLs.

```
aws wafv2 list-available-managed-rule-groups \
  --scope REGIONAL
```

Saída:

```
{
  "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."
    }
  ]
}

```

Para obter mais informações, consulte [Grupos de regras gerenciados](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListAvailableManagedRuleGroups](#) na Referência de AWS CLI Comandos.

list-ip-sets

O código de exemplo a seguir mostra como usar `list-ip-sets`.

AWS CLI

Para recuperar uma lista de conjuntos de IP

O seguinte `list-ip-sets` recupera todos os conjuntos de IP da conta que têm escopo regional.

```

aws wafv2 list-ip-sets \
  --scope REGIONAL

```

Saída:

```

{
  "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"
  }

```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListIpSets](#) na Referência de AWS CLI Comandos.

list-logging-configurations

O código de exemplo a seguir mostra como usar `list-logging-configurations`.

AWS CLI

Para recuperar uma lista de todas as configurações de registro de uma região

A seguir, são `list-logging-configurations` recuperadas todas as configurações de registro para a web ACLs que têm como escopo o uso regional na região. `us-west-2`

```

aws wafv2 list-logging-configurations \
  --scope REGIONAL \
  --region us-west-2

```

Saída:

```

{
  "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"
    ]
  }
]
}

```

Para obter mais informações, consulte [Registro de informações de ACL tráfego da Web](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListLoggingConfigurations](#) na Referência de AWS CLI Comandos.

list-regex-pattern-sets

O código de exemplo a seguir mostra como usar `list-regex-pattern-sets`.

AWS CLI

Para recuperar uma lista de conjuntos de padrões regex

O seguinte `list-regex-pattern-sets` recupera todos os conjuntos de padrões de regex para a conta que estão definidos na região. `us-west-2`

```

aws wafv2 list-regex-pattern-sets \
--scope REGIONAL \
--region us-west-2

```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListRegexPatternSets](#) na Referência de AWS CLI Comandos.

list-resources-for-web-acl

O código de exemplo a seguir mostra como usar `list-resources-for-web-acl`.

AWS CLI

Para recuperar os recursos associados a uma web ACL

O seguinte `list-resources-for-web-acl` recupera os REST API recursos do API Gateway que estão atualmente associados à web especificada ACL na região `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
```

Saída:

```
{
  "ResourceArns": [
    "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
```

```
]
}
```

Para obter mais informações, consulte [Associando ou desassociando uma Web ACL a um AWS recurso](#) no AWS Firewall AWS WAF Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListResourcesForWebAcl](#) na Referência de AWS CLI Comandos.

list-rule-groups

O código de exemplo a seguir mostra como usar `list-rule-groups`.

AWS CLI

Para recuperar uma lista de grupos de regras personalizados

O texto a seguir `list-rule-groups` recupera todos os grupos de regras personalizadas definidos para a conta no escopo e na região especificados.

```
aws wafv2 list-rule-groups \
  --scope REGIONAL \
  --region us-west-2
```

Saída:

```
{
  "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"
}

```

Para obter mais informações, consulte [Gerenciando seus próprios grupos de regras](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListRuleGroups](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para recuperar todas as tags de um recurso AWS WAF

O seguinte `list-tags-for-resource` recupera a lista de todos os pares de tag, chave e valor para a web ACL especificada.

```

aws wafv2 list-tags-for-resource \
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Saída:

```

{
  "NextMarker": "",
  "TagInfoForResource": {
    "ResourceARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
testwebacl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "TagList": [

    ]
  }
}

```

Para obter mais informações, consulte [Introdução ao AWS WAF AWS](#) Firewall Manager e ao AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-web-acls

O código de exemplo a seguir mostra como usar `list-web-acls`.

AWS CLI

Para recuperar a web ACLs para um escopo

O seguinte `list-web-acls` recupera toda a web ACLs que está definida para a conta no escopo especificado.

```
aws wafv2 list-web-acls \  
  --scope REGIONAL
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

Para obter mais informações, consulte [Gerenciando e usando uma lista de controle de acesso à Web \(WebACL\)](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [ListWebAcls](#) na Referência de AWS CLI Comandos.

put-logging-configuration

O código de exemplo a seguir mostra como usar `put-logging-configuration`.

AWS CLI

Para adicionar uma configuração de registro em uma web ACL

O seguinte `put-logging-configuration` adiciona a configuração de `aws-waf-logs-custom-transformation` registro do Amazon Kinesis Data Firehose à web ACL especificada, sem campos retirados dos registros.

```
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
```

Saída:

```
{  
  "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"  
    ]  
  }  
}
```

Para obter mais informações, consulte [Registro de informações de ACL tráfego da Web](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [PutLoggingConfiguration](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar `tag-resource`.

AWS CLI

Para adicionar tags a um AWS WAF recurso

O `tag-resource` exemplo a seguir adiciona uma tag com uma chave Name e um valor definidos AWSWAF para a web especificadaACL.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte [Introdução ao AWS WAF AWS Firewall Manager](#) e ao [AWS WAF AWS Shield Advanced Developer Guide](#).

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar `untag-resource`.

AWS CLI

Para remover tags de um AWS WAF recurso

O `untag-resource` exemplo a seguir remove a tag com a chave KeyName da web especificadaACL.

```
aws wafv2 untag-resource \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tag-keys "KeyName"
```

Este comando não produz saída.

Para obter mais informações, consulte [Introdução ao AWS WAF AWS Firewall Manager](#) e ao [AWS WAF AWS Shield Advanced Developer Guide](#).

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-ip-set

O código de exemplo a seguir mostra como usar `update-ip-set`.

AWS CLI

Para modificar as configurações de um conjunto IP existente

O seguinte `update-ip-set` atualiza as configurações do conjunto de IP especificado. Essa chamada requer um ID, que você pode obter da chamada `list-ip-sets`, e um token de bloqueio que você pode obter das chamadas `list-ip-sets get-ip-set` e. Essa chamada também retorna um token de bloqueio que você pode usar para uma atualização posterior.

```
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
```

Saída:

```
{  
  "NextLockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc"  
}
```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateIpSet](#) na Referência de AWS CLI Comandos.

update-regex-pattern-set

O código de exemplo a seguir mostra como usar `update-regex-pattern-set`.

AWS CLI

Para modificar as configurações de um conjunto de padrões regex existente

O seguinte `update-regex-pattern-set` atualiza as configurações do conjunto de padrões regex especificado. Essa chamada requer um ID, que você pode obter da chamada `list-regex-pattern-sets`, e um token de bloqueio que você pode obter das chamadas `list-regex-pattern-sets get-regex-pattern-set` e. Essa chamada também retorna um token de bloqueio que você pode usar para uma atualização posterior.


```
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
```

Saída:

```
{
  "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"
}
```

Para obter mais informações, consulte Conjuntos de [IP e conjuntos de padrões Regex](#) no AWS Firewall Manager e no AWS WAF AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateRegexPatternSet](#) na Referência de AWS CLI Comandos.

update-rule-group

O código de exemplo a seguir mostra como usar `update-rule-group`.

AWS CLI

Para atualizar um grupo de regras personalizado

O seguinte `update-rule-group` altera a configuração de visibilidade de um grupo de regras personalizadas existente. Essa chamada requer um ID, que você pode obter da chamada `list-rule-groups`, e um token de bloqueio que você pode obter das chamadas `list-rule-groups` `get-rule-group` e. Essa chamada também retorna um token de bloqueio que você pode usar para uma atualização posterior.

```
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
```

Saída:

```
{  
  "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"  
}
```

Para obter mais informações, consulte [Gerenciando seus próprios grupos de regras](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateRuleGroup](#) na Referência de AWS CLI Comandos.

update-web-acl

O código de exemplo a seguir mostra como usar `update-web-acl`.

AWS CLI

Para atualizar uma web ACL

O seguinte `update-web-acl` altera as configurações de uma web existente ACL. Essa chamada requer uma ID, que você pode obter da chamada `list-web-acls`, e um token de bloqueio e outras configurações, que você pode obter da chamada `get-web-acl`. Essa chamada também retorna um token de bloqueio que você pode usar para uma atualização posterior.

```
aws wafv2 update-web-acl \  
  --name TestWebAcl \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token 2294b3a1-0000-0000-0000-a3ae04329de9 \  
  --default-action Block={} \  
  --visibility-  
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=NewMetricTestW  
 \  
  --rules file://waf-rule.json \  
  --region us-west-2
```

Saída:

```
{  
  "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"
```

```
}
```

Para obter mais informações, consulte [Gerenciando e usando uma lista de controle de acesso à Web \(WebACL\)](#) no AWS WAF AWS Firewall Manager e no AWS Shield Advanced Developer Guide.

- Para API obter detalhes, consulte [UpdateWebAcl](#) na Referência de AWS CLI Comandos.

WorkDocs Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon WorkDocs.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

abort-document-version-upload

O código de exemplo a seguir mostra como usar `abort-document-version-upload`.

AWS CLI

Para interromper o upload de uma versão do documento

Este exemplo interrompe o upload de uma versão de documento iniciada anteriormente.

Comando:

```
aws workdocs abort-document-version-upload --document-id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2 --version-id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dccccd417da9313
```

Saída:

```
None
```

- Para API obter detalhes, consulte [AbortDocumentVersionUpload](#) na Referência de AWS CLI Comandos.

activate-user

O código de exemplo a seguir mostra como usar activate-user.

AWS CLI

Para ativar um usuário

Este exemplo ativa um usuário inativo.

Comando:

```
aws workdocs activate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Saída:

```
{
  "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"  
    }  
  }  
}  
}
```

- Para API obter detalhes, consulte [ActivateUser](#) na Referência de AWS CLI Comandos.

add-resource-permissions

O código de exemplo a seguir mostra como usar `add-resource-permissions`.

AWS CLI

Para adicionar permissões para um recurso

Este exemplo adiciona permissões ao recurso para os diretores especificados.

Comando:

```
aws workdocs add-resource-permissions --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --  
principals Id=anonymous, Type=ANONYMOUS, Role=VIEWER
```

Saída:

```
{  
  "ShareResults": [  
    {  
      "PrincipalId": "anonymous",  
      "Role": "VIEWER",  
      "Status": "SUCCESS",  
      "ShareId":  
"d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",  
      "StatusMessage": ""  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [AddResourcePermissions](#) na Referência de AWS CLI Comandos.

create-comment

O código de exemplo a seguir mostra como usar `create-comment`.

AWS CLI

Para adicionar um novo comentário

Este exemplo adiciona um novo comentário à versão especificada do documento.

Comando:

```
aws workdocs create-comment --document-  
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-  
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --  
text "This is a comment."
```

Saída:

```
{  
  "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"  
  }  
}
```

- Para API obter detalhes, consulte [CreateComment](#) na Referência de AWS CLI Comandos.

create-custom-metadata

O código de exemplo a seguir mostra como usar `create-custom-metadata`.

AWS CLI

Para criar metadados personalizados

Este exemplo cria metadados personalizados para o documento especificado.

Comando:

```
aws workdocs create-custom-metadata --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-metadata KeyName1=example,KeyName2=example2
```

Saída:

```
None
```

- Para API obter detalhes, consulte [CreateCustomMetadata](#) na Referência de AWS CLI Comandos.

create-folder

O código de exemplo a seguir mostra como usar `create-folder`.

AWS CLI

Para criar uma pasta

Este exemplo cria uma pasta.

Comando:

```
aws workdocs create-folder --name documents --parent-folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Saída:

```
{
  "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
  }
}
```

- Para API obter detalhes, consulte [CreateFolder](#) na Referência de AWS CLI Comandos.

create-labels

O código de exemplo a seguir mostra como usar `create-labels`.

AWS CLI

Para criar rótulos

Este exemplo cria uma série de etiquetas para um documento.

Comando:

```
aws workdocs create-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples" "my_documents"
```

Saída:

```
None
```

- Para API obter detalhes, consulte [CreateLabels](#) na Referência de AWS CLI Comandos.

create-notification-subscription

O código de exemplo a seguir mostra como usar `create-notification-subscription`.

AWS CLI

Para criar uma assinatura de notificação

O `create-notification-subscription` exemplo a seguir configura uma assinatura de notificação para a WorkDocs organização especificada da Amazon.

```
aws workdocs create-notification-subscription \  
  --organization-id d-123456789c \  
  --protocol HTTPS \  
  --subscription-type ALL \  
  --notification-endpoint "https://example.com/example"
```

Saída:

```
{  
  "Subscription": {  
    "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",  
    "EndPoint": "https://example.com/example",  
    "Protocol": "HTTPS"  
  }  
}
```

Para obter mais informações, consulte [Inscrever-se para receber notificações](#) no Amazon WorkDocs Developer Guide.

- Para API obter detalhes, consulte [CreateNotificationSubscription](#) na Referência de AWS CLI Comandos.

create-user

O código de exemplo a seguir mostra como usar `create-user`.

AWS CLI

Para criar um novo usuário

Este exemplo cria um novo usuário em um diretório Simple AD ou 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
```

Saída:

```
{
  "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"
      }
    }
  }
}
```

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

deactivate-user

O código de exemplo a seguir mostra como usar deactivate-user.

AWS CLI

Para desativar um usuário

Este exemplo desativa um usuário ativo.

Comando:

```
aws workdocs deactivate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeactivateUser](#) na Referência de AWS CLI Comandos.

delete-comment

O código de exemplo a seguir mostra como usar delete-comment.

AWS CLI

Para excluir um comentário especificado de uma versão do documento

Este exemplo exclui o comentário especificado da versão do documento especificada.

Comando:

```
aws workdocs delete-comment --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --comment-id 1534799058197-c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteComment](#) na Referência de AWS CLI Comandos.

delete-custom-metadata

O código de exemplo a seguir mostra como usar delete-custom-metadata.

AWS CLI

Para excluir metadados personalizados de um recurso

Este exemplo exclui todos os metadados personalizados do recurso especificado.

Comando:

```
aws workdocs delete-custom-metadata --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteCustomMetadata](#) na Referência de AWS CLI Comandos.

delete-document

O código de exemplo a seguir mostra como usar delete-document.

AWS CLI

Para excluir um documento

Este exemplo exclui o documento especificado.

Comando:

```
aws workdocs delete-document --document-id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteDocument](#) na Referência de AWS CLI Comandos.

delete-folder-contents

O código de exemplo a seguir mostra como usar `delete-folder-contents`.

AWS CLI

Para excluir o conteúdo de uma pasta

Este exemplo exclui o conteúdo da pasta especificada.

Comando:

```
aws workdocs delete-folder-contents --folder-  
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteFolderContents](#) na Referência de AWS CLI Comandos.

delete-folder

O código de exemplo a seguir mostra como usar `delete-folder`.

AWS CLI

Para excluir uma pasta

Este exemplo exclui a pasta especificada.

Comando:

```
aws workdocs delete-folder --folder-  
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteFolder](#) na Referência de AWS CLI Comandos.

delete-labels

O código de exemplo a seguir mostra como usar delete-labels.

AWS CLI

Para excluir rótulos

Este exemplo exclui os rótulos especificados de um documento.

Comando:

```
aws workdocs delete-labels --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --labels "documents" "examples"
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteLabels](#) na Referência de AWS CLI Comandos.

delete-notification-subscription

O código de exemplo a seguir mostra como usar delete-notification-subscription.

AWS CLI

Para excluir uma assinatura de notificação

O delete-notification-subscription exemplo a seguir exclui a assinatura de notificação especificada.

```
aws workdocs delete-notification-subscription \
  --subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \
  --organization-id d-123456789c
```

Este comando não produz saída.

Para obter mais informações, consulte [Inscrever-se para receber notificações](#) no Amazon WorkDocs Developer Guide.

- Para API obter detalhes, consulte [DeleteNotificationSubscription](#) na Referência de AWS CLI Comandos.

delete-user

O código de exemplo a seguir mostra como usar `delete-user`.

AWS CLI

Como excluir um usuário

Este exemplo exclui um usuário.

Comando:

```
aws workdocs delete-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Saída:

```
None
```

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

describe-activities

O código de exemplo a seguir mostra como usar `describe-activities`.

AWS CLI

Para obter uma lista das atividades do usuário

Este exemplo retorna uma lista das atividades mais recentes do usuário para a organização especificada, com um limite definido para as duas últimas atividades.

Comando:

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

Saída:

```
{
  "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
  }

```

- Para API obter detalhes, consulte [DescribeActivities](#) na Referência de AWS CLI Comandos.

describe-comments

O código de exemplo a seguir mostra como usar `describe-comments`.

AWS CLI

Para listar todos os comentários de uma versão específica do documento

Este exemplo lista todos os comentários da versão especificada do documento.

Comando:

```

aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920

```

Saída:

```

{
  "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"
    }
  ]
}

```

```
}
```

- Para API obter detalhes, consulte [DescribeComments](#) na Referência de AWS CLI Comandos.

describe-document-versions

O código de exemplo a seguir mostra como usar `describe-document-versions`.

AWS CLI

Para recuperar as versões de um documento

Este exemplo recupera as versões do documento especificado, incluindo as versões inicializadas e uma URL para o documento de origem.

Comando:

```
aws workdocs describe-document-versions --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

Saída:

```
{
  "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=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
    }
  },
  {
    "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=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
    }
  }
]
}

```

- Para API obter detalhes, consulte [DescribeDocumentVersions](#) na Referência de AWS CLI Comandos.

describe-folder-contents

O código de exemplo a seguir mostra como usar `describe-folder-contents`.

AWS CLI

Para descrever o conteúdo de uma pasta

Este exemplo descreve todo o conteúdo ativo da pasta especificada, incluindo seus documentos e subpastas, classificados por data em ordem crescente.

Comando:

```
aws workdocs describe-folder-contents --folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --order ASCENDING --type ALL
```

Saída:

```
{
  "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"
}
]
}

```

- Para API obter detalhes, consulte [DescribeFolderContents](#) na Referência de AWS CLI Comandos.

describe-groups

O código de exemplo a seguir mostra como usar `describe-groups`.

AWS CLI

Para recuperar uma lista de grupos

O `describe-groups` exemplo a seguir lista os grupos associados à WorkDocs organização especificada da Amazon.

```

aws workdocs describe-groups \
  --search-query "e" \
  --organization-id d-123456789c

```

Saída:

```

{
  "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"
      }
    ]
  }
```

Para obter mais informações, consulte [Getting Started with Amazon WorkDocs](#) no Guia de WorkDocs administração da Amazon.

- Para API obter detalhes, consulte [DescribeGroups](#) na Referência de AWS CLI Comandos.

describe-notification-subscriptions

O código de exemplo a seguir mostra como usar `describe-notification-subscriptions`.

AWS CLI

Para recuperar uma lista de assinaturas de notificação

O `describe-notification-subscriptions` exemplo a seguir recupera as assinaturas de notificação para a organização especificada da Amazon. WorkDocs

```
aws workdocs describe-notification-subscriptions \
  --organization-id d-123456789c
```

Saída:

```
{
  "Subscriptions": [
    {
      "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
      "EndPoint": "https://example.com/example",
      "Protocol": "HTTPS"
    }
  ]
}
```

Para obter mais informações, consulte [Inscrever-se para receber notificações](#) no Amazon WorkDocs Developer Guide.

- Para API obter detalhes, consulte [DescribeNotificationSubscriptions](#) na Referência de AWS CLI Comandos.

describe-resource-permissions

O código de exemplo a seguir mostra como usar `describe-resource-permissions`.

AWS CLI

Para obter uma lista de permissões para um recurso

Este exemplo retorna uma lista das permissões para o recurso especificado (documento ou pasta).

Comando:

```
aws workdocs describe-resource-permissions --resource-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

Saída:

```
{
  "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"
      }
    ]
  }
]
```

- Para API obter detalhes, consulte [DescribeResourcePermissions](#) na Referência de AWS CLI Comandos.

describe-users

O código de exemplo a seguir mostra como usar `describe-users`.

AWS CLI

Para recuperar detalhes de usuários especificados

Este exemplo recupera detalhes de todos os usuários na organização especificada.

Comando:

```
aws workdocs describe-users --organization-id d-926726012c
```

Saída:

```
{
  "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
    }
  ]
}

```

- Para API obter detalhes, consulte [DescribeUsers](#) na Referência de AWS CLI Comandos.

get-document-path

O código de exemplo a seguir mostra como usar `get-document-path`.

AWS CLI

Para recuperar as informações do caminho de um documento

Este exemplo recupera as informações do caminho (hierarquia da pasta raiz) do documento especificado e inclui os nomes das pastas principais.

Comando:

```
aws workdocs get-document-path --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME
```

Saída:

```
{
  "Path": {
    "Components": [
```

```

    {
      "Id":
"a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
      "Name": "/"
    },
    {
      "Id":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
      "Name": "Top Level Folder"
    },
    {
      "Id":
"d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "Name": "exampleDoc.docx"
    }
  ]
}

```

- Para API obter detalhes, consulte [GetDocumentPath](#) na Referência de AWS CLI Comandos.

get-document-version

O código de exemplo a seguir mostra como usar `get-document-version`.

AWS CLI

Para recuperar os metadados da versão de um documento especificado

Este exemplo recupera os metadados da versão do documento especificado, incluindo uma fonte URL e metadados personalizados.

Comando:

```

aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata

```

Saída:

```

{
  "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/15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3/1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920?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"
    }
}
}
}

```

- Para API obter detalhes, consulte [GetDocumentVersion](#) na Referência de AWS CLI Comandos.

get-document

O código de exemplo a seguir mostra como usar `get-document`.

AWS CLI

Para recuperar detalhes do documento

Este exemplo recupera os detalhes do documento especificado.

Comando:

```
aws workdocs get-document --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

Saída:

```
{
```

```

"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"
}
}

```

- Para API obter detalhes, consulte [GetDocument](#) na Referência de AWS CLI Comandos.

get-folder-path

O código de exemplo a seguir mostra como usar `get-folder-path`.

AWS CLI

Para recuperar informações de caminho para uma pasta

Este exemplo recupera as informações do caminho (hierarquia da pasta raiz) da pasta especificada e inclui os nomes das pastas principais.

Comando:

```
aws workdocs get-folder-path --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

Saída:

```
{
  "Path": {
    "Components": [
      {
        "Id":
"a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
        "Name": "/"
      },
      {
        "Id":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
"50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
        "Name": "Sublevel Folder"
      }
    ]
  }
}
```

- Para API obter detalhes, consulte [GetFolderPath](#) na Referência de AWS CLI Comandos.

get-folder

O código de exemplo a seguir mostra como usar `get-folder`.

AWS CLI

Para recuperar os metadados de uma pasta

Este exemplo recupera os metadados da pasta especificada.

Comando:

```
aws workdocs get-folder --folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Saída:

```
{
```

```
"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
}
}
```

- Para API obter detalhes, consulte [GetFolder](#) na Referência de AWS CLI Comandos.

get-resources

O código de exemplo a seguir mostra como usar `get-resources`.

AWS CLI

Para recuperar recursos compartilhados

O `get-resources` exemplo a seguir recupera os recursos compartilhados com o WorkDocs usuário especificado da Amazon.

```
aws workdocs get-resources \
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \
  --collection-type SHARED_WITH_ME
```

Saída:

```
{
  "Folders": [],
  "Documents": []
}
```

Para obter mais informações, consulte [Compartilhamento de arquivos e pastas](#) no Guia WorkDocs do usuário da Amazon.

- Para API obter detalhes, consulte [GetResources](#) na Referência de AWS CLI Comandos.

initiate-document-version-upload

O código de exemplo a seguir mostra como usar `initiate-document-version-upload`.

AWS CLI

Para iniciar o upload de uma versão do documento

O `initiate-document-upload` exemplo a seguir cria um novo objeto de documento e um objeto de versão.

```
aws workdocs initiate-document-version-upload \  
  --name exampledocname \  
  --parent-folder-  
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

Saída:

```
{  
  "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"
    }
}
}
}

```

- Para API obter detalhes, consulte [InitiateDocumentVersionUpload](#) na Referência de AWS CLI Comandos.

remove-all-resource-permissions

O código de exemplo a seguir mostra como usar `remove-all-resource-permissions`.

AWS CLI

Para remover todas as permissões de um recurso especificado

Este exemplo remove todas as permissões do recurso especificado.

Comando:

```
aws workdocs remove-all-resource-permissions --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Saída:

```
None
```

- Para API obter detalhes, consulte [RemoveAllResourcePermissions](#) na Referência de AWS CLI Comandos.

remove-resource-permission

O código de exemplo a seguir mostra como usar `remove-resource-permission`.

AWS CLI

Para remover permissões de um recurso

Este exemplo remove as permissões do recurso para o principal especificado.

Comando:

```
aws workdocs remove-resource-permission --resource-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-id anonymous
```

Saída:

```
None
```

- Para API obter detalhes, consulte [RemoveResourcePermission](#) na Referência de AWS CLI Comandos.

update-document-version

O código de exemplo a seguir mostra como usar `update-document-version`.

AWS CLI

Para alterar o status da versão de um documento para Ativo

Este exemplo altera o status da versão do documento para Ativo.

Comando:

```
aws workdocs update-document-version --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --version-status ACTIVE
```

Saída:

None

- Para API obter detalhes, consulte [UpdateDocumentVersion](#) na Referência de AWS CLI Comandos.

update-document

O código de exemplo a seguir mostra como usar `update-document`.

AWS CLI

Para atualizar um documento

Este exemplo atualiza o nome e a pasta principal de um documento.

Comando:

```
aws workdocs update-document --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --name updatedDoc --parent-folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Saída:

None

- Para API obter detalhes, consulte [UpdateDocument](#) na Referência de AWS CLI Comandos.

update-folder

O código de exemplo a seguir mostra como usar `update-folder`.

AWS CLI

Para atualizar uma pasta

Este exemplo atualiza o nome e a pasta principal de uma pasta.

Comando:

```
aws workdocs update-folder --folder-  
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --  
name exampleFolder1 --parent-folder-  
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Saída:

```
None
```

- Para API obter detalhes, consulte [UpdateFolder](#) na Referência de AWS CLI Comandos.

update-user

O código de exemplo a seguir mostra como usar update-user.

AWS CLI

Para atualizar um usuário

Este exemplo atualiza o fuso horário do usuário especificado.

Comando:

```
aws workdocs update-user --user-  
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-  
id "America/Los_Angeles"
```

Saída:

```
{  
  "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"
  }
}
}
```

- Para API obter detalhes, consulte [UpdateUser](#) na Referência de AWS CLI Comandos.

WorkMail Exemplos da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface com a Amazon WorkMail.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

associate-delegate-to-resource

O código de exemplo a seguir mostra como usar `associate-delegate-to-resource`.

AWS CLI

Para adicionar um delegado a um recurso

O `associate-delegate-to-resource` comando a seguir adiciona um delegado a um recurso.

```
aws workmail associate-delegate-to-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AssociateDelegateToResource](#) na Referência de AWS CLI Comandos.

associate-member-to-group

O código de exemplo a seguir mostra como usar `associate-member-to-group`.

AWS CLI

Para adicionar um membro a um grupo

O `associate-member-to-group` comando a seguir adiciona o membro especificado a um grupo.

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [AssociateMemberToGroup](#) na Referência de AWS CLI Comandos.

create-alias

O código de exemplo a seguir mostra como usar `create-alias`.

AWS CLI

Para criar um alias

O `create-alias` comando a seguir cria um alias para a entidade especificada (usuário ou grupo).

```
aws workmail create-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

Este comando não produz saída.

- Para API obter detalhes, consulte [CreateAlias](#) na Referência de AWS CLI Comandos.

create-group

O código de exemplo a seguir mostra como usar `create-group`.

AWS CLI

Para criar um novo grupo

O `create-group` comando a seguir cria um novo grupo para a organização especificada.

```
aws workmail create-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleGroup1
```

Saída:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-333334444-4444"  
}
```

- Para API obter detalhes, consulte [CreateGroup](#) na Referência de AWS CLI Comandos.

create-resource

O código de exemplo a seguir mostra como usar `create-resource`.

AWS CLI

Para criar um novo recurso

O `create-resource` comando a seguir cria um novo recurso (sala de reuniões) para a organização especificada.

```
aws workmail create-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleRoom1 \  
  --type ROOM
```

Saída:

```
{  
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"  
}
```

- Para API obter detalhes, consulte [CreateResource](#) na Referência de AWS CLI Comandos.

create-user

O código de exemplo a seguir mostra como usar create-user.

AWS CLI

Para criar um novo usuário

O create-user comando a seguir cria um novo usuário.

```
aws workmail create-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleName \  
  --display-name exampleDisplayName \  
  --password examplePa$$w0rd
```

Saída:

```
{  
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"  
}
```

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

delete-access-control-rule

O código de exemplo a seguir mostra como usar delete-access-control-rule.

AWS CLI

Para excluir uma regra de controle de acesso

O `delete-access-control-rule` exemplo a seguir exclui a regra de controle de acesso especificada da WorkMail organização especificada da Amazon.

```
aws workmail delete-access-control-rule \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --name "myRule"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com regras de controle de acesso](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [DeleteAccessControlRule](#) na Referência de AWS CLI Comandos.

`delete-alias`

O código de exemplo a seguir mostra como usar `delete-alias`.

AWS CLI

Para excluir um alias

O `delete-alias` comando a seguir exclui o alias da entidade especificada (usuário ou grupo).

```
aws workmail delete-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteAlias](#) na Referência de AWS CLI Comandos.

`delete-group`

O código de exemplo a seguir mostra como usar `delete-group`.

AWS CLI

Para excluir um grupo existente

O `delete-group` comando a seguir exclui um grupo existente da Amazon WorkMail.

```
aws workmail delete-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteGroup](#) na Referência de AWS CLI Comandos.

`delete-mailbox-permissions`

O código de exemplo a seguir mostra como usar `delete-mailbox-permissions`.

AWS CLI

Para excluir permissões de caixa de correio

O `delete-mailbox-permissions` comando a seguir exclui as permissões de caixa de correio que foram concedidas anteriormente a um usuário ou grupo. A entidade representa o usuário que possui a caixa de correio, e o beneficiário representa o usuário ou grupo para quem excluir as permissões.

```
aws workmail delete-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --grantee-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteMailboxPermissions](#) na Referência de AWS CLI Comandos.

`delete-resource`

O código de exemplo a seguir mostra como usar `delete-resource`.

AWS CLI

Para excluir um recurso existente

O `delete-resource` comando a seguir exclui um recurso existente da Amazon WorkMail.

```
aws workmail delete-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteResource](#) na Referência de AWS CLI Comandos.

`delete-user`

O código de exemplo a seguir mostra como usar `delete-user`.

AWS CLI

Como excluir um usuário

O `delete-user` comando a seguir exclui o usuário especificado da Amazon WorkMail e de todos os sistemas subsequentes.

```
aws workmail delete-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

`deregister-from-work-mail`

O código de exemplo a seguir mostra como usar `deregister-from-work-mail`.

AWS CLI

Para desativar uma entidade existente

O `deregister-from-work-mail` comando a seguir impede que uma entidade existente (usuário, grupo ou recurso) use a Amazon WorkMail.

```
aws workmail deregister-from-work-mail \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DeregisterFromWorkMail](#) na Referência de AWS CLI Comandos.

describe-group

O código de exemplo a seguir mostra como usar `describe-group`.

AWS CLI

Para recuperar informações de um grupo

O `describe-group` comando a seguir recupera informações sobre o grupo especificado.

```
aws workmail describe-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Saída:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",  
  "Name": "exampleGroup1",  
  "State": "ENABLED"  
}
```

- Para API obter detalhes, consulte [DescribeGroup](#) na Referência de AWS CLI Comandos.

describe-organization

O código de exemplo a seguir mostra como usar `describe-organization`.

AWS CLI

Para recuperar informações de uma organização

O `describe-organization` comando a seguir recupera informações da WorkMail organização especificada da Amazon.

```
aws workmail describe-organization \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Saída:

```
{  
  "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"  
}
```

Para obter mais informações, consulte [Working with Organizations](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [DescribeOrganization](#) na Referência de AWS CLI Comandos.

describe-resource

O código de exemplo a seguir mostra como usar `describe-resource`.

AWS CLI

Para recuperar informações de um recurso

O `describe-resource` comando a seguir recupera informações sobre o recurso especificado.

```
aws workmail describe-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c
```

Saída:

```
{
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
  "Name": "exampleRoom1",
  "Type": "ROOM",
  "BookingOptions": {
    "AutoAcceptRequests": true,
    "AutoDeclineRecurringRequests": false,
    "AutoDeclineConflictingRequests": true
  },
  "State": "ENABLED"
}
```

- Para API obter detalhes, consulte [DescribeResource](#) na Referência de AWS CLI Comandos.

describe-user

O código de exemplo a seguir mostra como usar `describe-user`.

AWS CLI

Para recuperar informações do usuário

O `describe-user` comando a seguir recupera informações sobre o usuário especificado.

```
aws workmail describe-user \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Saída:

```
{
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",
  "Name": "exampleUser1",
  "Email": "exampleUser1@site.awsapps.com",
  "DisplayName": "",
  "State": "ENABLED",
  "UserRole": "USER",
  "EnabledDate": 1532459261.827
}
```

- Para API obter detalhes, consulte [DescribeUser](#) na Referência de AWS CLI Comandos.

disassociate-delegate-from-resource

O código de exemplo a seguir mostra como usar `disassociate-delegate-from-resource`.

AWS CLI

Para remover um membro de um recurso

O `disassociate-delegate-from-resource` comando a seguir remove o membro especificado de um recurso.

```
aws workmail disassociate-delegate-from-resource \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DisassociateDelegateFromResource](#) na Referência de AWS CLI Comandos.

disassociate-member-from-group

O código de exemplo a seguir mostra como usar `disassociate-member-from-group`.

AWS CLI

Para remover um membro de um grupo

O `disassociate-member-from-group` comando a seguir remove o membro especificado de um grupo.

```
aws workmail disassociate-member-from-group \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --group-id S-1-1-11-1122222222-2222233333-333334444-4444 \
  --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Este comando não produz saída.

- Para API obter detalhes, consulte [DisassociateMemberFromGroup](#) na Referência de AWS CLI Comandos.

get-access-control-effect

O código de exemplo a seguir mostra como usar `get-access-control-effect`.

AWS CLI

Para obter o efeito das regras de controle de acesso

O `get-access-control-effect` exemplo a seguir recupera o efeito das regras de controle de acesso da WorkMail organização Amazon especificada para o endereço IP, a ação do protocolo de acesso e o ID do usuário especificados.

```
aws workmail get-access-control-effect \
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
  --ip-address "192.0.2.0" \
  --action "WindowsOutlook" \
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333"
```

Saída:

```
{
  "Effect": "DENY",
  "MatchedRules": [
    "myRule"
  ]
}
```

Para obter mais informações, consulte Como [trabalhar com regras de controle de acesso](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [GetAccessControlEffect](#) na Referência de AWS CLI Comandos.

get-mailbox-details

O código de exemplo a seguir mostra como usar `get-mailbox-details`.

AWS CLI

Para obter os detalhes da caixa de correio de um usuário

O `get-mailbox-details` comando a seguir recupera detalhes sobre a caixa de correio do usuário especificado.

```
aws workmail get-mailbox-details \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Saída:

```
{  
  "MailboxQuota": 51200,  
  "MailboxSize": 0.03890800476074219  
}
```

Para obter mais informações, consulte [Gerenciamento de contas de usuário](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [GetMailboxDetails](#) na Referência de AWS CLI Comandos.

list-access-control-rules

O código de exemplo a seguir mostra como usar `list-access-control-rules`.

AWS CLI

Para listar as regras de controle de acesso

O `list-access-control-rules` exemplo a seguir lista as regras de controle de acesso para a WorkMail organização especificada da Amazon.

```
aws workmail list-access-control-rules \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

Saída:

```
{  
  "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
  }
]
}

```

Para obter mais informações, consulte Como [trabalhar com regras de controle de acesso](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [ListAccessControlRules](#) na Referência de AWS CLI Comandos.

list-aliases

O código de exemplo a seguir mostra como usar `list-aliases`.

AWS CLI

Para listar aliases para um membro

O `list-aliases` comando a seguir lista aliases para o membro especificado (usuário ou grupo).

```

aws workmail list-aliases \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333

```

Saída:

```

{
  "Aliases": [

```

```
    "exampleAlias@site.awsapps.com",  
    "exampleAlias1@site.awsapps.com"  
  ]  
}
```

- Para API obter detalhes, consulte [ListAliases](#) na Referência de AWS CLI Comandos.

list-group-members

O código de exemplo a seguir mostra como usar `list-group-members`.

AWS CLI

Para listar membros do grupo

O `list-group-members` comando a seguir lista os membros do grupo especificado.

```
aws workmail list-group-members \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Saída:

```
{  
  "Members": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Name": "exampleUser1",  
      "Type": "USER",  
      "State": "ENABLED",  
      "EnabledDate": 1532459261.827  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListGroupMembers](#) na Referência de AWS CLI Comandos.

list-groups

O código de exemplo a seguir mostra como usar `list-groups`.

AWS CLI

Para recuperar uma lista de grupos

O `list-groups` comando a seguir recupera resumos dos grupos na organização especificada.

```
aws workmail list-groups \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Saída:

```
{  
  "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"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListGroups](#) na Referência de AWS CLI Comandos.

`list-mailbox-permissions`

O código de exemplo a seguir mostra como usar `list-mailbox-permissions`.

AWS CLI

Para recuperar permissões de caixa de correio

O `list-mailbox-permissions` comando a seguir recupera as permissões de caixa de correio associadas à caixa de correio da entidade especificada.

```
aws workmail list-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Saída:

```
{
  "Permissions": [
    {
      "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
      "GranteeType": "USER",
      "PermissionValues": [
        "FULL_ACCESS"
      ]
    }
  ]
}
```

- Para API obter detalhes, consulte [ListMailboxPermissions](#) na Referência de AWS CLI Comandos.

list-organizations

O código de exemplo a seguir mostra como usar `list-organizations`.

AWS CLI

Para recuperar uma lista de organizações

O `list-organizations` comando a seguir recupera resumos de organizações não excluídas.

```
aws workmail list-organizations
```

Saída:

```
{
  "OrganizationSummaries": [
    {
      "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
      "Alias": "exampleAlias",
      "State": "Active"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListOrganizations](#) na Referência de AWS CLI Comandos.

list-resource-delegates

O código de exemplo a seguir mostra como usar `list-resource-delegates`.

AWS CLI

Para listar os delegados de um recurso

O `list-resource-delegates` comando a seguir recupera os delegados associados ao recurso especificado.

```
aws workmail list-resource-delegates \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

Saída:

```
{  
  "Delegates": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Type": "USER"  
    }  
  ]  
}
```

- Para API obter detalhes, consulte [ListResourceDelegates](#) na Referência de AWS CLI Comandos.

list-resources

O código de exemplo a seguir mostra como usar `list-resources`.

AWS CLI

Para recuperar uma lista de recursos

O `list-resources` comando a seguir recupera resumos dos recursos da organização especificada.

```
aws workmail list-resources \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Saída:

```
{
  "Resources": [
    {
      "Id": "r-7afe0efbade843a58cdc10251fce992c",
      "Name": "exampleRoom1",
      "Type": "ROOM",
      "State": "ENABLED"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListResources](#) na Referência de AWS CLI Comandos.

list-tags-for-resource

O código de exemplo a seguir mostra como usar `list-tags-for-resource`.

AWS CLI

Para listar as tags de um recurso

O `list-tags-for-resource` exemplo a seguir lista as tags da WorkMail organização especificada da Amazon.

```
aws workmail list-tags-for-resource \
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
  n1pq2345678r901st2u3vx45x6789yza
```

Saída:

```
{
  "Tags": [
    {
      "Key": "priority",
      "Value": "1"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte Como [marcar uma organização](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [ListTagsForResource](#) na Referência de AWS CLI Comandos.

list-users

O código de exemplo a seguir mostra como usar `list-users`.

AWS CLI

Para recuperar uma lista de usuários

O `list-users` comando a seguir recupera resumos dos usuários na organização especificada.

```
aws workmail list-users \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Saída:

```
{
  "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"
    }
  ]
}
```

- Para API obter detalhes, consulte [ListUsers](#) na Referência de AWS CLI Comandos.

put-access-control-rule

O código de exemplo a seguir mostra como usar `put-access-control-rule`.

AWS CLI

Para colocar uma nova regra de controle de acesso

O `put-access-control-rule` exemplo a seguir nega ao usuário especificado o acesso à WorkMail organização especificada da Amazon.

```
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
```

Este comando não produz saída.

Para obter mais informações, consulte Como [trabalhar com regras de controle de acesso](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [PutAccessControlRule](#) na Referência de AWS CLI Comandos.

put-mailbox-permissions

O código de exemplo a seguir mostra como usar `put-mailbox-permissions`.

AWS CLI

Para definir permissões de caixa de correio

O `put-mailbox-permissions` comando a seguir define permissões de acesso total para o beneficiário especificado (usuário ou grupo). A entidade representa o proprietário da caixa de correio.

```
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 \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```



```
--permission-values FULL_ACCESS
```

Este comando não produz saída.

- Para API obter detalhes, consulte [PutMailboxPermissions](#) na Referência de AWS CLI Comandos.

register-to-work-mail

O código de exemplo a seguir mostra como usar `register-to-work-mail`.

AWS CLI

Para registrar uma entidade existente ou desativada

O `register-to-work-mail` comando a seguir permite que a entidade existente especificada (usuário, grupo ou recurso) use a 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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [RegisterToWorkMail](#) na Referência de AWS CLI Comandos.

reset-password

O código de exemplo a seguir mostra como usar `reset-password`.

AWS CLI

Para redefinir a senha de um usuário

O `reset-password` comando a seguir redefine a senha do usuário especificado.

```
aws workmail reset-password \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --password examplePa$$w0rd
```

Este comando não produz saída.

- Para API obter detalhes, consulte [ResetPassword](#) na Referência de AWS CLI Comandos.

tag-resource

O código de exemplo a seguir mostra como usar tag-resource.

AWS CLI

Para aplicar uma tag a um recurso

O tag-resource exemplo a seguir aplica uma tag com chave “prioridade” e valor “1” à WorkMail organização especificada da Amazon.

```
aws workmail tag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tags "Key=priority,Value=1"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar uma organização](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [TagResource](#) na Referência de AWS CLI Comandos.

untag-resource

O código de exemplo a seguir mostra como usar untag-resource.

AWS CLI

Para desmarcar um recurso

O untag-resource exemplo a seguir remove a tag especificada da WorkMail organização especificada da Amazon.

```
aws workmail untag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tag-key Key=priority
```

```
--tag-keys "priority"
```

Este comando não produz saída.

Para obter mais informações, consulte Como [marcar uma organização](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [UntagResource](#) na Referência de AWS CLI Comandos.

update-mailbox-quota

O código de exemplo a seguir mostra como usar update-mailbox-quota.

AWS CLI

Para atualizar a cota da caixa de correio de um usuário

O update-mailbox-quota comando a seguir altera a cota da caixa de correio do usuário especificado.

```
aws workmail update-mailbox-quota \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --mailbox-quota 40000
```

Este comando não produz saída.

Para obter mais informações, consulte [Gerenciamento de contas de usuário](#) no Amazon WorkMail Administrator Guide.

- Para API obter detalhes, consulte [UpdateMailboxQuota](#) na Referência de AWS CLI Comandos.

update-primary-email-address

O código de exemplo a seguir mostra como usar update-primary-email-address.

AWS CLI

Para atualizar um endereço de e-mail principal

O update-primary-email-address comando a seguir atualiza o endereço de e-mail principal da entidade especificada (usuário, grupo ou recurso).

```
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
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdatePrimaryEmailAddress](#) na Referência de AWS CLI Comandos.

update-resource

O código de exemplo a seguir mostra como usar `update-resource`.

AWS CLI

Para atualizar um recurso

O `update-resource` comando a seguir atualiza o nome do recurso especificado.

```
aws workmail update-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c \  
  --name exampleRoom2
```

Este comando não produz saída.

- Para API obter detalhes, consulte [UpdateResource](#) na Referência de AWS CLI Comandos.

Exemplos de fluxo de WorkMail mensagens da Amazon usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface Amazon WorkMail Message Flow.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

get-raw-message-content

O código de exemplo a seguir mostra como usar `get-raw-message-content`.

AWS CLI

Para obter o conteúdo bruto de uma mensagem de e-mail

O `get-raw-message-content` exemplo a seguir obtém o conteúdo bruto de uma mensagem de e-mail em trânsito e a envia para um arquivo de texto chamado `test`.

```
aws workmailmessageflow get-raw-message-content \  
  --message-id a1b2cd34-ef5g-6h7j-kl8m-npq9012345rs \  
  test
```

Conteúdo do arquivo `test` após a execução do 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+--
```

Para obter mais informações, consulte [Recuperando o conteúdo da mensagem com o AWS Lambda](#) no WorkMail Amazon Administrator Guide.

- Para API obter detalhes, consulte [GetRawMessageContent](#) na Referência de AWS CLI Comandos.

WorkSpaces exemplos usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with WorkSpaces.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

create-tags

O código de exemplo a seguir mostra como usar `create-tags`.

AWS CLI

Para adicionar tags a um Workspace

O `create-tags` exemplo a seguir adiciona as tags especificadas às especificadas Workspace.

```
aws workspaces create-tags \  
  --resource-id ws-dk1xzr417 \  
  --tags Key=Department,Value=Finance
```

Este comando não produz saída.

Para obter mais informações, consulte [WorkSpaces Recursos de tags](#) no Guia de WorkSpaces administração da Amazon.

- Para API obter detalhes, consulte [CreateTags](#) na Referência de AWS CLI Comandos.

create-workspaces

O código de exemplo a seguir mostra como usar `create-workspaces`.

AWS CLI

Exemplo 1: Para criar um AlwaysOn Workspace

O `create-workspaces` exemplo a seguir cria um AlwaysOn Workspace para o usuário especificado, usando o diretório e o pacote especificados.

```
aws workspaces create-workspaces \  
  --workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

Saída:

```
{  
  "FailedRequests": [],  
  "PendingRequests": [  
    {
```

```

        "WorkspaceId": "ws-kcqms853t",
        "DirectoryId": "d-926722edaf",
        "UserName": "Mateo",
        "State": "PENDING",
        "BundleId": "wsb-0zsvgp8fc"
    }
]
}

```

Exemplo 2: Para criar um AutoStop WorkSpace

O `create-workspaces` exemplo a seguir cria um AutoStop WorkSpace para o usuário especificado, usando o diretório e o pacote especificados.

```

aws workspaces create-workspaces \
  --
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties

```

Saída:

```

{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-dk1x zr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "State": "PENDING",
      "BundleId": "wsb-0zsvgp8fc"
    }
  ]
}

```

Exemplo 3: Para criar um usuário desacoplado WorkSpace

O `create-workspaces` exemplo a seguir cria um usuário desacoplado WorkSpace definindo o nome de usuário como e especificando um WorkSpace nome[UNDEFINED], ID do diretório e ID do pacote.

```

aws workspaces create-workspaces \
  --workspaces
  DirectoryId=d-926722edaf,UserName=''[UNDEFINED]'' ,WorkspaceName=MaryWorkspace1,BundleId=wsb

```


Saída:

```
{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-abcd1234",
      "DirectoryId": "d-926722edaf",
      "UserName": "[UNDEFINED]",
      "State": "PENDING",
      "BundleId": "wsb-0zsvgp8fc",
      "WorkspaceName": "MaryWorkspace1"
    }
  ]
}
```

Para obter mais informações, consulte [Iniciar um desktop virtual](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [CreateWorkspaces](#) na Referência de AWS CLI Comandos.

delete-tags

O código de exemplo a seguir mostra como usar delete-tags.

AWS CLI

Para excluir uma tag de um Workspace

O delete-tags exemplo a seguir exclui a tag especificada da especificada Workspace.

```
aws workspaces delete-tags \
  --resource-id ws-dk1x zr417 \
  --tag-keys Department
```

Este comando não produz saída.

Para obter mais informações, consulte [WorkSpaces Recursos de tags](#) no Guia de WorkSpaces administração da Amazon.

- Para API obter detalhes, consulte [DeleteTags](#) na Referência de AWS CLI Comandos.

deregister-workspace-directory

O código de exemplo a seguir mostra como usar `deregister-workspace-directory`.

AWS CLI

Para cancelar o registro de um diretório

O `deregister-workspace-directory` exemplo a seguir cancela o registro do diretório especificado.

```
aws workspaces deregister-workspace-directory \  
  --directory-id d-926722edaf
```

Este comando não produz saída.

Para obter mais informações, consulte [Registrar um diretório WorkSpaces](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [DeregisterWorkspaceDirectory](#) na Referência de AWS CLI Comandos.

describe-tags

O código de exemplo a seguir mostra como usar `describe-tags`.

AWS CLI

Para descrever as tags de um WorkSpace

O `describe-tags` exemplo a seguir descreve as tags para o especificado WorkSpace.

```
aws workspaces describe-tags \  
  --resource-id ws-dk1xzz417
```

Saída:

```
{  
  "TagList": [  
    {  
      "Key": "Department",  
      "Value": "Finance"  
    }  
  ]  
}
```

```
]
}
```

Para obter mais informações, consulte [WorkSpaces Recursos de tags](#) no Guia de WorkSpaces administração da Amazon.

- Para API obter detalhes, consulte [DescribeTags](#) na Referência de AWS CLI Comandos.

describe-workspace-bundles

O código de exemplo a seguir mostra como usar `describe-workspace-bundles`.

AWS CLI

Para listar os pacotes fornecidos pela Amazon

O `describe-workspace-bundles` exemplo a seguir lista os nomes e IDs os pacotes fornecidos pela Amazon, em formato de tabela e classificados por nome.

```
aws workspaces describe-workspace-bundles \
  --owner AMAZON \
  --query "Bundles[*].[Name, BundleId]"
```

Saída:

```
[
  [
    "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"  
  ],  
  ...  
]
```

Para obter mais informações, consulte [WorkSpaces pacotes e imagens](#) no Guia de WorkSpaces administração da Amazon.

- Para API obter detalhes, consulte [DescribeWorkspaceBundles](#) na Referência de AWS CLI Comandos.

describe-workspace-directories

O código de exemplo a seguir mostra como usar `describe-workspace-directories`.

AWS CLI

Para descrever um diretório registrado

O `describe-workspace-directories` exemplo a seguir descreve o diretório registrado especificado.

```
aws workspaces describe-workspace-directories \  
  --directory-ids d-926722edaf
```

Saída:

```
{  
  "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"
    }
  }
]
}

```

Para obter mais informações, consulte [Gerenciar diretórios WorkSpaces](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [DescribeWorkspaceDirectories](#) na Referência de AWS CLI Comandos.

describe-workspaces-connection-status

O código de exemplo a seguir mostra como usar `describe-workspaces-connection-status`.

AWS CLI

Para descrever o status da conexão de um Workspace

O `describe-workspaces-connection-status` exemplo a seguir descreve o status da conexão especificada Workspace.

```
aws workspaces describe-workspaces-connection-status \  
  --workspace-ids ws-dk1x zr417
```

Saída:

```
{  
  "WorkspacesConnectionStatus": [  
    {  
      "WorkspaceId": "ws-dk1x zr417",  
      "ConnectionState": "CONNECTED",  
      "ConnectionStateCheckTimestamp": 1662526214.744  
    }  
  ]  
}
```

Para obter mais informações, consulte [Administrar seu WorkSpaces](#) no Guia de WorkSpaces Administração da Amazon.

- Para API obter detalhes, consulte [DescribeWorkspacesConnectionStatus](#) na Referência de AWS CLI Comandos.

`describe-workspaces`

O código de exemplo a seguir mostra como usar `describe-workspaces`.

AWS CLI

Para descrever um Workspace

O `describe-workspaces` exemplo a seguir descreve o especificado Workspace.

```
aws workspaces describe-workspaces \  
  --workspace-ids ws-dk1x zr417
```

Saída:

```
{
  "Workspaces": [
    {
      "WorkspaceId": "ws-dk1x zr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "IpAddress": "172.16.0.175",
      "State": "STOPPED",
      "BundleId": "wsb-0zsvgp8fc",
      "SubnetId": "subnet-500d5819",
      "ComputerName": "WSAMZN-RBSLTTD9",
      "WorkspaceProperties": {
        "RunningMode": "AUTO_STOP",
        "RunningModeAutoStopTimeoutInMinutes": 60,
        "RootVolumeSizeGib": 80,
        "UserVolumeSizeGib": 10,
        "ComputeTypeName": "VALUE"
      },
      "ModificationStates": []
    }
  ]
}
```

Para obter mais informações, consulte [Administrar seu WorkSpaces](#) no Guia de WorkSpaces Administração da Amazon.

- Para API obter detalhes, consulte [DescribeWorkspaces](#) na Referência de AWS CLI Comandos.

migrate-workspace

O código de exemplo a seguir mostra como usar migrate-workspace.

AWS CLI

Para migrar um WorkSpace

O migrate-workspace exemplo a seguir migra o pacote especificado WorkSpace para o especificado.

```
aws workspaces migrate-workspace \
  --source-workspace-id ws-dk1x zr417 \
```

```
--bundle-id wsb-j4dky1gs4
```

Saída:

```
{
  "SourceWorkspaceId": "ws-dk1xzr417",
  "TargetWorkspaceId": "ws-x5h11bkp5"
}
```

Para obter mais informações, consulte [Migrate a WorkSpace](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [MigrateWorkSpace](#) na Referência de AWS CLI Comandos.

modify-workspace-creation-properties

O código de exemplo a seguir mostra como usar `modify-workspace-creation-properties`.

AWS CLI

Para modificar uma propriedade de WorkSpace criação de um diretório

O `modify-workspace-creation-properties` exemplo a seguir ativa a `EnableInternetAccess` propriedade para o diretório especificado. Isso permite a atribuição automática de endereços IP públicos para o WorkSpaces criado para o diretório.

```
aws workspaces modify-workspace-creation-properties \  
  --resource-id d-926722edaf \  
  --workspace-creation-properties EnableInternetAccess=true
```

Este comando não produz saída.

Para obter mais informações, consulte [Atualizar os detalhes do seu diretório WorkSpaces](#) no Guia de WorkSpaces Administração da Amazon.

- Para API obter detalhes, consulte [ModifyWorkspaceCreationProperties](#) na Referência de AWS CLI Comandos.

modify-workspace-properties

O código de exemplo a seguir mostra como usar `modify-workspace-properties`.

AWS CLI

Para modificar o modo de execução de um Workspace

O `modify-workspace-properties` exemplo a seguir define o modo de execução do especificado Workspace para `AUTO_STOP`.

```
aws workspaces modify-workspace-properties \  
  --workspace-id ws-dk1xzz417 \  
  --workspace-properties RunningMode=AUTO_STOP
```

Este comando não produz saída.

Para obter mais informações, consulte [Modificar a Workspace](#) no Guia de WorkSpaces Administração da Amazon.

- Para API obter detalhes, consulte [ModifyWorkspaceProperties](#) na Referência de AWS CLI Comandos.

`modify-workspace-state`

O código de exemplo a seguir mostra como usar `modify-workspace-state`.

AWS CLI

Para modificar o estado de um Workspace

O `modify-workspace-state` exemplo a seguir define o estado do especificado Workspace para `ADMIN_MAINTENANCE`.

```
aws workspaces modify-workspace-state \  
  --workspace-id ws-dk1xzz417 \  
  --workspace-state ADMIN_MAINTENANCE
```

Este comando não produz saída.

Para obter mais informações, consulte a [Workspace manutenção](#) no Guia de WorkSpaces Administração da Amazon.

- Para API obter detalhes, consulte [ModifyWorkspaceState](#) na Referência de AWS CLI Comandos.

reboot-workspaces

O código de exemplo a seguir mostra como usar `reboot-workspaces`.

AWS CLI

Para reinicializar um Workspace

O `reboot-workspaces` exemplo a seguir reinicializa o especificado Workspace.

```
aws workspaces reboot-workspaces \  
  --reboot-workspace-requests ws-dk1xzr417
```

Saída:

```
{  
  "FailedRequests": []  
}
```

Para obter mais informações, consulte [Reboot a Workspace](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [RebootWorkspaces](#) na Referência de AWS CLI Comandos.

rebuild-workspaces

O código de exemplo a seguir mostra como usar `rebuild-workspaces`.

AWS CLI

Para reconstruir um Workspace

O `rebuild-workspaces` exemplo a seguir reconstrói o especificado Workspace

```
aws workspaces rebuild-workspaces \  
  --rebuild-workspace-requests ws-dk1xzr417
```

Saída:

```
{  
  "FailedRequests": []  
}
```

```
}
```

Para obter mais informações, consulte [Rebuild a WorkSpace](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [RebuildWorkspaces](#) na Referência de AWS CLI Comandos.

register-workspace-directory

O código de exemplo a seguir mostra como usar `register-workspace-directory`.

AWS CLI

Para registrar um diretório

O `register-workspace-directory` exemplo a seguir registra o diretório especificado para uso com a Amazon WorkSpaces.

```
aws workspaces register-workspace-directory \  
  --directory-id d-926722edaf \  
  --no-enable-work-docs
```

Este comando não produz saída.

Para obter mais informações, consulte [Registrar um diretório WorkSpaces](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [RegisterWorkspaceDirectory](#) na Referência de AWS CLI Comandos.

restore-workspace

O código de exemplo a seguir mostra como usar `restore-workspace`.

AWS CLI

Para restaurar um WorkSpace

O `restore-workspace` exemplo a seguir restaura o especificado WorkSpace.

```
aws workspaces restore-workspace \  
  --workspace-id ws-926722edaf
```

```
--workspace-id ws-dk1xzt417
```

Este comando não produz saída.

Para obter mais informações, consulte [Restore a Workspace](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [RestoreWorkspaces](#) na Referência de AWS CLI Comandos.

start-workspaces

O código de exemplo a seguir mostra como usar start-workspaces.

AWS CLI

Para iniciar um AutoStop Workspace

O start-workspaces exemplo a seguir inicia o especificado Workspace. Eles Workspace devem ter um modo de execução deAutoStop.

```
aws workspaces start-workspaces \  
  --start-workspace-requests WorkspaceId=ws-dk1xzt417
```

Saída:

```
{  
  "FailedRequests": []  
}
```

Para obter mais informações, consulte [Stop and start an AutoStop Workspace](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [StartWorkspaces](#) na Referência de AWS CLI Comandos.

stop-workspaces

O código de exemplo a seguir mostra como usar stop-workspaces.

AWS CLI

Para parar um AutoStop Workspace

O `stop-workspaces` exemplo a seguir interrompe o especificado WorkSpace. Eles WorkSpace devem ter um modo de execução de `AutoStop`.

```
aws workspaces stop-workspaces \  
  --stop-workspace-requests WorkspaceId=ws-dk1xzz417
```

Saída:

```
{  
  "FailedRequests": []  
}
```

Para obter mais informações, consulte [Stop and start an AutoStop WorkSpace](#) no Amazon WorkSpaces Administration Guide.

- Para API obter detalhes, consulte [StopWorkspaces](#) na Referência de AWS CLI Comandos.

terminate-workspaces

O código de exemplo a seguir mostra como usar `terminate-workspaces`.

AWS CLI

Para encerrar um WorkSpace

O `terminate-workspaces` exemplo a seguir encerra o espaço de trabalho especificado.

```
aws workspaces terminate-workspaces \  
  --terminate-workspace-requests ws-dk1xzz417
```

Saída:

```
{  
  "FailedRequests": []  
}
```

Para obter mais informações, consulte [Excluir um WorkSpace](#) no Guia de WorkSpaces Administração da Amazon.

- Para API obter detalhes, consulte [TerminateWorkspaces](#) na Referência de AWS CLI Comandos.

Exemplos de raio-X usando AWS CLI

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o AWS Command Line Interface with X-Ray.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

batch-traces-get

O código de exemplo a seguir mostra como usar `batch-traces-get`.

AWS CLI

Para obter uma lista de traços

O `batch-get-traces` exemplo a seguir recupera uma lista de rastreamentos especificados por um ID. O rastreamento completo inclui um documento para cada segmento, compilado a partir de todos os documentos de segmento recebidos com a mesma ID de rastreamento.

```
aws xray batch-get-traces \  
  --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

Saída:

```
{  
  "Traces": [  
    {  
      "Id": "1-5d82881a-0a9126e92a73e971eed891b9",  
      "Duration": 0.232,  
    }  
  ]  
}
```

```

    "Segments": [
      {
        "Id": "54aff5735b12dd28",
        "Document": "{\"id\":\"54aff5735b12dd28\",\"name\":
\\\"Scorekeep\\\",\\\"start_time\\\":1.568835610432E9,\\\"end_time\\\":1.568835610664E9,
\\\"http\\\":{\\\"request\\\":{\\\"url\\\":\\\"http://scorekeep-env-1.m4fg2pfzpv.us-
east-2.elasticbeanstalk.com/api/user\\\",\\\"method\\\":\\\"POST\\\",\\\"user_agent\\\":
\\\"curl/7.59.0\\\",\\\"client_ip\\\":\\\"52.95.4.28\\\",\\\"x_forwarded_for\\\":true},
\\\"response\\\":{\\\"status\\\":200}},\\\"aws\\\":{\\\"elastic_beanstalk\\\":{\\\"version_label
\\\":\\\"Sample Application-1\\\",\\\"deployment_id\\\":3,\\\"environment_name\\\":\\\"Scorekeep-
env-1\\\"},\\\"ec2\\\":{\\\"availability_zone\\\":\\\"us-east-2b\\\",\\\"instance_id\\\":
\\\"i-0e3cf4d2de0f3f37a\\\"},\\\"xray\\\":{\\\"sdk_version\\\":\\\"1.1.0\\\",\\\"sdk\\\":\\\"X-Ray for
Java\\\"}},\\\"service\\\":{\\\"runtime\\\":\\\"OpenJDK 64-Bit Server VM\\\",\\\"runtime_version
\\\":\\\"1.8.0_222\\\"},\\\"trace_id\\\":\\\"1-5d82881a-0a9126e92a73e971eed891b9\\\",
\\\"origin\\\":\\\"AWS::ElasticBeanstalk::Environment\\\",\\\"subsegments\\\":[{\\\"id\\\":
\\\"2d6900034ccfe558\\\",\\\"name\\\":\\\"DynamoDB\\\",\\\"start_time\\\":1.568835610658E9,
\\\"end_time\\\":1.568835610664E9,\\\"http\\\":{\\\"response\\\":{\\\"status\\\":200,
\\\"content_length\\\":61}},\\\"aws\\\":{\\\"table_name\\\":\\\"scorekeep-user\\\",\\\"operation\\\":
\\\"UpdateItem\\\",\\\"request_id\\\":\\\"TPEIDNDUROMLPOV17U4A79555Nvv4KQNS05AEMVJF66Q9ASUAAJG
\\\",\\\"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\\\":
\\\"TPEIDNDUROMLPOV17U4A79555Nvv4KQNS05AEMVJF66Q9ASUAAJG\\\",\\\"resource_names\\\":
[\\\"scorekeep-user\\\"]},\\\"trace_id\\\":\\\"1-5d82881a-0a9126e92a73e971eed891b9\\\",\\\"origin
\\\":\\\"AWS::DynamoDB::Table\\\"}"
      }
    ],
    "UnprocessedTraceIds": []
  }

```

Para obter mais informações, consulte [Usando o AWS X-Ray API com o AWS CLI](#) no AWS X-Ray Developer Guide.

- Para API obter detalhes, consulte [BatchTracesGet](#) na Referência de AWS CLI Comandos.

create-group

O código de exemplo a seguir mostra como usar `create-group`.

AWS CLI

Para criar um grupo

O `create-group` exemplo a seguir cria um recurso de grupo chamado `AdminGroup`. O grupo obtém uma expressão de filtro que define os critérios do grupo como um segmento relacionado a um serviço específico que causa uma falha ou um erro.

```
aws xray create-group \  
  --group-name "AdminGroup" \  
  --filter-expression "service(\"mydomain.com\") {fault OR error}"
```

Saída:

```
{  
  "GroupName": "AdminGroup",  
  "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",  
  "FilterExpression": "service(\"mydomain.com\") {fault OR error}"  
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [CreateGroup](#) na Referência de AWS CLI Comandos.

create-sampling-rule

O código de exemplo a seguir mostra como usar `create-sampling-rule`.

AWS CLI

Para criar uma regra de amostragem

O `create-sampling-rule` exemplo a seguir cria uma regra para controlar o comportamento de amostragem para aplicativos instrumentados. As regras são fornecidas por um JSON arquivo. A maioria dos campos da regra de amostragem é necessária para criar a regra.

```
aws xray create-sampling-rule \  
  --rule-name "MySamplingRule" \  
  --rule-definition '{"RuleName": "MySamplingRule", "FilterExpression": "service(\"mydomain.com\") {fault OR error}"}
```



```
--cli-input-json file://9000-base-scorekeep.json
```

Conteúdo de 9000-base-scorekeep.json:

```
{
  "SamplingRule": {
    "RuleName": "base-scorekeep",
    "ResourceARN": "*",
    "Priority": 9000,
    "FixedRate": 0.1,
    "ReservoirSize": 5,
    "ServiceName": "Scorekeep",
    "ServiceType": "*",
    "Host": "*",
    "HTTPMethod": "*",
    "URLPath": "*",
    "Version": 1
  }
}
```

Saída:

```
{
  "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
  }
}
```

```
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWSAWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [CreateSamplingRule](#) na Referência de AWS CLI Comandos.

delete-group

O código de exemplo a seguir mostra como usar `delete-group`.

AWS CLI

Para excluir um grupo

O `delete-group` exemplo a seguir exclui o recurso de grupo especificado.

```
aws xray delete-group \  
  --group-name "AdminGroup" \  
  --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

Este comando não produz saída.

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWSAWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [DeleteGroup](#) na Referência de AWS CLI Comandos.

delete-sampling-rule

O código de exemplo a seguir mostra como usar `delete-sampling-rule`.

AWS CLI

Para excluir uma regra de amostragem

O `delete-sampling-rule` exemplo a seguir exclui a regra de amostragem especificada. Você pode especificar o grupo usando o nome do grupo ou o grupoARN.

```
aws xray delete-sampling-rule \  
  --rule-name polling-scorekeep
```

Saída:

```
{
  "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
  }
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [DeleteSamplingRule](#) na Referência de AWS CLI Comandos.

get-encryption-config

O código de exemplo a seguir mostra como usar `get-encryption-config`.

AWS CLI

Para recuperar a configuração de criptografia

O `get-encryption-config` exemplo a seguir recupera a configuração de criptografia atual dos dados do AWS X-Ray.

```
aws xray get-encryption-config
```

Saída:

```
{
  "EncryptionConfig": {
    "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
    "Status": "ACTIVE",
    "Type": "NONE"
  }
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [GetEncryptionConfig](#) na Referência de AWS CLI Comandos.

get-group

O código de exemplo a seguir mostra como usar `get-group`.

AWS CLI

Para recuperar um grupo

O `get-group` exemplo a seguir exibe detalhes do recurso de grupo especificado. Os detalhes incluem o nome do grupo, o grupo ARN e a expressão do filtro que define os critérios desse grupo. Os grupos também podem ser recuperados por ARN.

```
aws xray get-group \
  --group-name "AdminGroup"
```

Saída:

```
{
  "Group": [
    {
      "GroupName": "AdminGroup",
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
      "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
    }
  ]
}
```

```
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWSAWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [GetGroup](#) na Referência de AWS CLI Comandos.

get-groups

O código de exemplo a seguir mostra como usar `get-groups`.

AWS CLI

Para recuperar todos os grupos

O exemplo a seguir exibe detalhes de todos os grupos ativos.

```
aws xray get-groups
```

Saída:

```
{
  "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"
    }
  ]
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWSAWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [GetGroups](#) na Referência de AWS CLI Comandos.

get-sampling-rules

O código de exemplo a seguir mostra como usar `get-sampling-rules`.

AWS CLI

Para recuperar todas as regras de amostragem

O `get-sampling-rules` exemplo a seguir exibe detalhes de todas as regras de amostragem disponíveis. :

```
aws xray get-sampling-rules
```

Saída:

```
{
  "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
}
]
}

```

Para obter mais informações, consulte [Usando regras de amostragem com o X-Ray API no X-Ray Developer Guide](#).AWS

- Para API obter detalhes, consulte [GetSamplingRules](#) na Referência de AWS CLI Comandos.

get-sampling-targets

O código de exemplo a seguir mostra como usar `get-sampling-targets`.

AWS CLI

Para solicitar uma cota de amostragem

O `get-sampling-targets` exemplo a seguir solicita uma cota de amostragem para as regras que o serviço está usando para amostras de solicitações. A resposta do AWS X-Ray inclui uma cota que pode ser usada em vez de ser emprestada do reservatório.

```
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 } ]'
```

Saída:

```
{
  "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": []
}
```

Para obter mais informações, consulte [Usando regras de amostragem com o X-Ray API no X-Ray Developer Guide](#).AWS

- Para API obter detalhes, consulte [GetSamplingTargets](#) na Referência de AWS CLI Comandos.

get-service-graph

O código de exemplo a seguir mostra como usar `get-service-graph`.

AWS CLI

Para obter um gráfico de serviços

O exemplo a seguir exibe um documento dentro de um período de tempo especificado que descreve os serviços que processam as solicitações recebidas e os serviços posteriores que eles chamam como resultado. :

```
aws xray get-service-graph \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

Saída:

```
{  
  "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  
}
```

Para obter mais informações, consulte [Usando o AWS X-Ray API com o AWS CLI](#) no AWS X-Ray Developer Guide.

- Para API obter detalhes, consulte [GetServiceGraph](#) na Referência de AWS CLI Comandos.

get-trace-summaries

O código de exemplo a seguir mostra como usar `get-trace-summaries`.

AWS CLI

Para obter um resumo do rastreamento

O `get-trace-summaries` exemplo a seguir recupera os IDs metadados dos rastreamentos disponíveis em um período de tempo especificado.

```
aws xray get-trace-summaries \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

Saída:

```
[  
  "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"  
]
```

Para obter mais informações, consulte [Usando o AWS X-Ray API com o AWS CLI](#) no AWS X-Ray Developer Guide.

- Para API obter detalhes, consulte [GetTraceSummaries](#) na Referência de AWS CLI Comandos.

put-encryption-config

O código de exemplo a seguir mostra como usar `put-encryption-config`.

AWS CLI

Para atualizar a configuração de criptografia

O seguinte `put-encryption-config` exemplo atualiza a configuração de criptografia para os dados do X-Ray do AWS para usar a chave gerenciada pelo AWS KMS padrão `aws/xray`.

```
aws xray put-encryption-config \  
  --type KMS \  
  --key-id alias/aws/xray
```

Saída:

```
{  
  "EncryptionConfig": {  
    "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-  
b0ea215cbba5",  
    "Status": "UPDATING",  
    "Type": "KMS"  
  }  
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [PutEncryptionConfig](#) na Referência de AWS CLI Comandos.

put-trace-segments

O código de exemplo a seguir mostra como usar `put-trace-segments`.

AWS CLI

Para fazer o upload de um segmento

O `put-trace-segments` exemplo a seguir carrega documentos do segmento para o AWS X-Ray. O documento do segmento é consumido como uma lista de documentos do 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}}, \"inferred\": true, \"aws\": {\"consistent_read\": false, \"table_name
  \": \"scorekeep-session-xray\", \"operation\": \"GetItem\", \"request_id\":
  \"SCAU230M6M8F038UASGC7785ARVV4KQNS05AEMVJF66Q9ASUAAJG\", \"resource_names\":
  [\"scorekeep-session-xray\"]}, \"origin\": \"AWS::DynamoDB::Table\"}"
```

Saída:

```
{
  "UnprocessedTraceSegments": []
}
```

Para obter mais informações, consulte [Enviando dados de rastreamento para o AWS X-Ray no AWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [PutTraceSegments](#) na Referência de AWS CLI Comandos.

update-group

O código de exemplo a seguir mostra como usar `update-group`.

AWS CLI

Para atualizar um grupo

O `update-group` exemplo a seguir atualiza os critérios pelos quais aceitar rastreamentos no grupo nomeado `AdminGroup`. Você pode especificar o grupo desejado usando o nome do grupo ou o grupo 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}"
```

Saída:

```
{
  "GroupName": "AdminGroup",
  "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
  "FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [UpdateGroup](#) na Referência de AWS CLI Comandos.

update-sampling-rule

O código de exemplo a seguir mostra como usar `update-sampling-rule`.

AWS CLI

Para atualizar uma regra de amostragem

O `update-sampling-rule` exemplo a seguir modifica a configuração de uma regra de amostragem. As regras são consumidas a partir de um JSON arquivo. Somente os campos que estão sendo atualizados são obrigatórios.

```
aws xray update-sampling-rule \
  --cli-input-json file://1000-default.json
```

Conteúdo de `1000-default.json`:

```
{
  "SamplingRuleUpdate": {
    "RuleName": "Default",
    "FixedRate": 0.01,
    "ReservoirSize": 0
  }
}
```

Saída:

```
{
  "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
}
]
```

Para obter mais informações, consulte [Definindo configurações de amostragem, grupos e criptografia com o X-Ray API no AWSAWS X-Ray Developer Guide](#).

- Para API obter detalhes, consulte [UpdateSamplingRule](#) na Referência de AWS CLI Comandos.

AWS CLI com exemplos de código de script Bash

Os exemplos de código neste tópico mostram como usar o script AWS Command Line Interface with Bash with AWS.

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Os cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Serviços

- [Exemplos AWS CLI do DynamoDB usando o script Bash](#)
- [EC2Exemplos da Amazon usando AWS CLI com o script Bash](#)
- [HealthImaging exemplos de uso AWS CLI com o script Bash](#)
- [IAMexemplos de uso AWS CLI com o script Bash](#)
- [Exemplos do Amazon S3 usando o script AWS CLI Bash](#)
- [AWS STS exemplos de uso AWS CLI com o script Bash](#)

Exemplos AWS CLI do DynamoDB usando o script Bash

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o script AWS Command Line Interface with Bash com o DynamoDB.

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Conceitos básicos](#)
- [Ações](#)

Conceitos básicos

Aprender os conceitos básicos

O exemplo de código a seguir mostra como:

- Criar uma tabela que possa conter dados de filmes.
- Colocar, obter e atualizar um único filme na tabela.
- Grave dados do filme na tabela a partir de um JSON arquivo de amostra.
- Consultar filmes que foram lançados em determinado ano.

- Verificar filmes que foram lançados em um intervalo de anos.
- Excluir um filme da tabela e, depois, excluir a tabela.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

O cenário de conceitos básicos do 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"
  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
}

```

As funções do DynamoDB usadas nesse cenário.

```

#####
# 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 getopt "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
}
```

```
#####
# 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

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 getopts "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}" ;;
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
}

```

As funções utilitárias usadas nesse cenário.

```
#####
```

```
# 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."
    fi
}
```

```
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
}
```


- Para API obter detalhes, consulte os tópicos a seguir na Referência de AWS CLI Comandos.
 - [BatchWriteItem](#)
 - [CreateTable](#)
 - [DeleteItem](#)
 - [DeleteTable](#)
 - [DescribeTable](#)
 - [GetItem](#)
 - [PutItem](#)
 - [Query](#)
 - [Scan](#)
 - [UpdateItem](#)

Ações

BatchGetItem

O código de exemplo a seguir mostra como usar BatchGetItem.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
        esac
    done
}
```

```

        ;;
    \?)
        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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}


```

- Para API obter detalhes, consulte [BatchGetItem](#) na Referência de AWS CLI Comandos.

BatchWriteItem

O código de exemplo a seguir mostra como usar BatchWriteItem.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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

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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [BatchWriteItem](#) na Referência de AWS CLI Comandos.

CreateTable

O código de exemplo a seguir mostra como usar CreateTable.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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."
fi
```

```
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
}
```

As funções utilitárias usadas neste exemplo.

```
#####
# 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."
    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
}

```

- Para API obter detalhes, consulte [CreateTable](#) na Referência de AWS CLI Comandos.

DeleteItem

O código de exemplo a seguir mostra como usar DeleteItem.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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 getopt "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
}

```

```

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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [Deleteltemna](#) Referência de AWS CLI Comandos.

DeleteTable

O código de exemplo a seguir mostra como usar DeleteTable.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
        esac
    done
}
```

```

        ;;
    \?)
        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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [DeleteTable](#) na Referência de AWS CLI Comandos.

DescribeTable

O código de exemplo a seguir mostra como usar DescribeTable.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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."
    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
}

```

- Para API obter detalhes, consulte [DescribeTable](#) na Referência de AWS CLI Comandos.

GetItem

O código de exemplo a seguir mostra como usar GetItem.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

```
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
}
```

As funções utilitárias usadas neste exemplo.

```
#####
# 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
}

```

- Para API obter detalhes, consulte [GetItem](#) na Referência de AWS CLI Comandos.

ListTables

O código de exemplo a seguir mostra como usar ListTables.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [ListTables](#) na Referência de AWS CLI Comandos.

PutItem

O código de exemplo a seguir mostra como usar PutItem.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}


```

- Para API obter detalhes, consulte [PutItem](#) na Referência de AWS CLI Comandos.

Query

O código de exemplo a seguir mostra como usar Query.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}
```

- Para API obter detalhes, consulte [Consulta](#) na Referência de AWS CLI Comandos.

Scan

O código de exemplo a seguir mostra como usar Scan.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

As funções utilitárias usadas neste exemplo.

```
#####
# 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
}

```

- Para API obter detalhes, consulte [Escanear](#) na Referência de AWS CLI Comandos.

UpdateItem

O código de exemplo a seguir mostra como usar UpdateItem.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}
```

- Para API obter detalhes, consulte [UpdateItem](#) na Referência de AWS CLI Comandos.

EC2 Exemplos da Amazon usando AWS CLI com o script Bash

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o script AWS Command Line Interface with Bash com a AmazonEC2.

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Conceitos básicos](#)
- [Ações](#)

Conceitos básicos

Aprender os conceitos básicos

O exemplo de código a seguir mostra como:

- Criar um par de chaves e um grupo de segurança.
- Selecione uma Amazon Machine Image (AMI) e um tipo de instância compatível e, em seguida, crie uma instância.

- Interromper e reiniciar a instância.
- Associar um endereço IP elástico à sua instância.
- Conecte-se à sua instância eSSH, em seguida, limpe os recursos.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

Execute um cenário interativo em um prompt de comando.

```
#####
# 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
```

```
fi

chmod 400 "${key_file_name}"

if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then
  local keys_and_fingerprints
  keys_and_fingerprints="$(ec2_describe_key_pairs)" && {
    local image_name_and_id
    while IFS=$'\n' read -r image_name_and_id; do
      local entries
      IFS=$'\t' read -ra entries <<<"$image_name_and_id"
      echo "Found rsa key ${entries[0]} with fingerprint:"
      echo "    ${entries[1]}"
    done <<<"$keys_and_fingerprints"
  }
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create a security group to manage access to your instance."
echo -n "Enter a unique name for your security group: "
get_input
local security_group_name
security_group_name=$get_input_result
local security_group_id
security_group_id=$(ec2_create_security_group -n "$security_group_name" \
  -d "Security group for EC2 instance") || {
  errecho "The security failed to create. This demo will exit."
  clean_up "$key_name" "$key_file_name"
  return 1
}

echo "Security group created with ID $security_group_id"
echo

local public_ip
public_ip=$(curl -s http://checkip.amazonaws.com)

echo "Let's add a rule to allow SSH only from your current IP address."
echo "Your public IP address is $public_ip"
echo -n "press return to add this rule to your security group."
get_input
```

```
if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i "$public_ip"
-p tcp -f 22 -t 22; then
    errecho "The security group rules failed to update. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo "Security group rules updated"

local security_group_description
security_group_description="$(ec2_describe_security_groups -g
"${security_group_id}")" || {
    errecho "Failed to describe security groups. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

mapfile -t parameters <<<"$security_group_description"
IFS=$'\t' read -ra entries <<<"${parameters[0]}"
echo "Security group: ${entries[0]}"
echo "    ID: ${entries[1]}"
echo "    VPC: ${entries[2]}"
echo "Inbound permissions:"
IFS=$'\t' read -ra entries <<<"${parameters[1]}"
echo "    IpProtocol: ${entries[0]}"
echo "    FromPort: ${entries[1]}"
echo "    ToPort: ${entries[2]}"
echo "    CidrIp: ${parameters[2]}"

local parameters
parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
    errecho "Failed to get parameters. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local image_ids=""
mapfile -t parameters <<<"$parameters"
for image_name_and_id in "${parameters[@]}"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
```

```
        image_ids+="${values[1]} "
    fi
done

local images
images="$(ec2_describe_images -i "$image_ids")" || {
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

new_line_and_tab_to_list "$images"
local images=("${list_result[@]}")

# Get the size of the array
local images_count=${#images[@]}

if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
fi

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some options:"
for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
done

integer_input "Please enter the number of the AMI you want to use: " 1
"$((images_count / 3))"
local choice=$get_input_result
choice=$((choice - 1) * 3)

echo "Great choice."
echo

local architecture=${images[$((choice + 1))]}
local image_id=${images[$((choice + 2))]}
echo "Here are some instance types that support the ${architecture} architecture
of the image:"
```

```
response="$(ec2_describe_instance_types -a "${architecture}" -t
"*.micro,*.small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

local instance_types
mapfile -t instance_types <<<"$response"

# Get the size of the array
local instance_types_count=${#instance_types[@]}

echo "Here are some options:"
for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
done

integer_input "Which one do you want to use? " 1 "${#instance_types[@]}"
"
choice=$get_input_result
local instance_type=${instance_types[$((choice - 1))]}
echo "Another great choice."
echo

echo "Creating your instance and waiting for it to start..."
local instance_id
instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k "$key_name"
-s "$security_group_id") || {
    errecho "Failed to run instance. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
}

ec2_wait_for_instance_running -i "$instance_id"
echo "Your instance is ready:"
echo

local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

echo
print_instance_details "${instance_details}"
```

```
local public_ip
public_ip=$(echo "${instance_details}" | awk '{print $6}')
echo
echo "You can use SSH to connect to your instance"
echo "If the connection attempt times out, you might have to manually update the
SSH ingress rule"
echo "for your IP address in the AWS Management Console."
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
ec2_wait_for_instance_running -i "$instance_id"

echo "Your instance is running again."
local instance_details
instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

public_ip=$(echo "${instance_details}" | awk '{print $6}')

echo "Every time your instance is restarted, its public IP address changes"
connect_to_instance "$key_file_name" "$public_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "You can allocate an Elastic IP address and associate it with your instance"
echo "to keep a consistent IP address even when your instance restarts."
```

```
local result
result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
}

local elastic_ip allocation_id
elastic_ip=$(echo "$result" | awk '{print $1}')
allocation_id=$(echo "$result" | awk '{print $2}')

echo "Allocated static Elastic IP address: $elastic_ip"

local association_id
association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id") || {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
"$allocation_id"
    return 1
}

echo "Associated your Elastic IP with your instance."
echo "You can now use SSH to connect to your instance by using the Elastic IP."
connect_to_instance "$key_file_name" "$elastic_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

echo "Let's stop and start your instance to see what changes."
echo "Stopping your instance and waiting until it's stopped..."
ec2_stop_instances -i "$instance_id"
ec2_wait_for_instance_stopped -i "$instance_id"

echo "Your instance is stopped. Restarting..."

ec2_start_instances -i "$instance_id"
ec2_wait_for_instance_running -i "$instance_id"

echo "Your instance is running again."
local instance_details
```



```

instance_details="$(ec2_describe_instances -i "${instance_id}")"

print_instance_details "${instance_details}"

echo "Because you have associated an Elastic IP with your instance, you can"
echo "connect by using a consistent IP address after the instance restarts."
connect_to_instance "$key_file_name" "$elastic_ip"

echo -n "Press Enter when you're ready to continue the demo: "
get_input

echo_repeat "*" 88
echo_repeat "*" 88

if yes_no_input "Do you want to delete the resources created in this demo: (y/n)"; then
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \
        "$allocation_id" "$association_id"
else
    echo "The following resources were not deleted."
    echo "Key pair: $key_name"
    echo "Key file: $key_file_name"
    echo "Security group: $security_group_id"
    echo "Instance: $instance_id"
    echo "Elastic IP address: $elastic_ip"
fi
}

#####
# function clean_up
#
# This function cleans up the created resources.
# $1 - The name of the ec2 key pair to delete.
# $2 - The name of the key file to delete.
# $3 - The ID of the security group to delete.
# $4 - The ID of the instance to terminate.
# $5 - The ID of the elastic IP address to release.
# $6 - The ID of the elastic IP address to disassociate.
#
# Returns:
# 0 - If successful.
# 1 - If an error occurred.
#####
function clean_up() {

```

```
local result=0
local key_pair_name=$1
local key_file_name=$2
local security_group_id=$3
local instance_id=$4
local allocation_id=$5
local association_id=$6

if [ -n "$association_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_disassociate_address -a "$association_id"); then
        echo "Disassociated elastic IP address with ID $association_id"
    else
        errecho "The elastic IP address disassociation failed."
        result=1
    fi
fi

if [ -n "$allocation_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_release_address -a "$allocation_id"); then
        echo "Released elastic IP address with ID $allocation_id"
    else
        errecho "The elastic IP address release failed."
        result=1
    fi
fi

if [ -n "$instance_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_terminate_instances -i "$instance_id"); then
        echo "Started terminating instance with ID $instance_id"

        ec2_wait_for_instance_terminated -i "$instance_id"
    else
        errecho "The instance terminate failed."
        result=1
    fi
fi

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
}

```

As funções do DynamoDB usadas nesse cenário.

```

#####
# 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
}

```

As funções utilitárias usadas nesse cenário.

```

#####
# 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
}
```

- Para API obter detalhes, consulte os tópicos a seguir na Referência de AWS CLI Comandos.
 - [AllocateAddress](#)
 - [AssociateAddress](#)
 - [AuthorizeSecurityGroupIngress](#)
 - [CreateKeyPair](#)
 - [CreateSecurityGroup](#)
 - [DeleteKeyPair](#)
 - [DeleteSecurityGroup](#)
 - [DescribeImages](#)
 - [DescribeInstanceTypes](#)

- [DescribeInstances](#)
- [DescribeKeyPairs](#)
- [DescribeSecurityGroups](#)
- [DisassociateAddress](#)
- [ReleaseAddress](#)
- [RunInstances](#)
- [StartInstances](#)
- [StopInstances](#)
- [TerminateInstances](#)
- [UnmonitorInstances](#)

Ações

AllocateAddress

O código de exemplo a seguir mostra como usar AllocateAddress.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [AllocateAddress](#) na Referência de AWS CLI Comandos.

AssociateAddress

O código de exemplo a seguir mostra como usar AssociateAddress.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [AssociateAddress](#) na Referência de AWS CLI Comandos.

AuthorizeSecurityGroupIngress

O código de exemplo a seguir mostra como usar AuthorizeSecurityGroupIngress.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [AuthorizeSecurityGroupIngress](#) na Referência de AWS CLI Comandos.

CreateKeyPair

O código de exemplo a seguir mostra como usar CreateKeyPair.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

```

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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [CreateKeyPair](#) na Referência de AWS CLI Comandos.

CreateSecurityGroup

O código de exemplo a seguir mostra como usar CreateSecurityGroup.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```



```

    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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}
```

- Para API obter detalhes, consulte [CreateSecurityGroup](#) na Referência de AWS CLI Comandos.

DeleteKeyPair

O código de exemplo a seguir mostra como usar DeleteKeyPair.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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"
```

```

        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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [DeleteKeyPair](#) na Referência de AWS CLI Comandos.

DeleteSecurityGroup

O código de exemplo a seguir mostra como usar DeleteSecurityGroup.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}
```

- Para API obter detalhes, consulte [DeleteSecurityGroup](#) na Referência de AWS CLI Comandos.

DescribeImages

O código de exemplo a seguir mostra como usar DescribeImages.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [DescreverImagens](#) na Referência de AWS CLI Comandos.

DescribeInstanceTypes

O código de exemplo a seguir mostra como usar DescribeInstanceTypes.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [DescribeInstanceTypes](#) na Referência de AWS CLI Comandos.

DescribeInstances

O código de exemplo a seguir mostra como usar DescribeInstances.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}
```

- Para API obter detalhes, consulte [DescribeInstances](#) na Referência de AWS CLI Comandos.

DescribeKeyPairs

O código de exemplo a seguir mostra como usar DescribeKeyPairs.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [DescribeKeyPairs](#) na Referência de AWS CLI Comandos.

DescribeSecurityGroups

O código de exemplo a seguir mostra como usar DescribeSecurityGroups.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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)."

```

```

    return 1
fi

echo "$response"

return 0
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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."
    fi
}

```

```
elif [ "$err_code" == 253 ]; then
    errecho " The system environment or configuration was invalid."
elif [ "$err_code" == 254 ]; then
    errecho " The service returned an error."
elif [ "$err_code" == 255 ]; then
    errecho " 255 is a catch-all error."
fi

return 0
}
```

- Para API obter detalhes, consulte [DescribeSecurityGroups](#) na Referência de AWS CLI Comandos.

DisassociateAddress

O código de exemplo a seguir mostra como usar DisassociateAddress.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}

```

- Para API obter detalhes, consulte [DisassociateAddress](#) na Referência de AWS CLI Comandos.

ReleaseAddress

O código de exemplo a seguir mostra como usar ReleaseAddress.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

As funções utilitárias usadas neste exemplo.

```
#####
# 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
}
```

- Para API obter detalhes, consulte [ReleaseAddress](#) na Referência de AWS CLI Comandos.

RunInstances

O código de exemplo a seguir mostra como usar RunInstances.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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  
}
```

- Para API obter detalhes, consulte [RunInstances](#) na Referência de AWS CLI Comandos.

StartInstances

O código de exemplo a seguir mostra como usar StartInstances.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}


```

- Para API obter detalhes, consulte [StartInstances](#) na Referência de AWS CLI Comandos.

StopInstances

O código de exemplo a seguir mostra como usar StopInstances.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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
}


```

- Para API obter detalhes, consulte [StopInstances](#) na Referência de AWS CLI Comandos.

TerminateInstances

O código de exemplo a seguir mostra como usar TerminateInstances.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

As funções utilitárias usadas neste exemplo.

```

#####
# 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  
}
```

- Para API obter detalhes, consulte [TerminateInstances](#) na Referência de AWS CLI Comandos.

HealthImaging exemplos de uso AWS CLI com o script Bash

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o script AWS Command Line Interface with Bash with HealthImaging.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

CreateDatastore

O código de exemplo a seguir mostra como usar CreateDatastore.

AWS CLI com 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
}

```

- Para API obter detalhes, consulte [CreateDatastore](#) na Referência de AWS CLI Comandos.

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

DeleteDatastore

O código de exemplo a seguir mostra como usar DeleteDatastore.

AWS CLI com 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
}

```

- Para API obter detalhes, consulte [DeleteDatastore](#) na Referência de AWS CLI Comandos.

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

GetDatastore

O código de exemplo a seguir mostra como usar GetDatastore.

AWS CLI com 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
}

```

- Para API obter detalhes, consulte [GetDatastore](#) na Referência de AWS CLI Comandos.

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

ListDatastores

O código de exemplo a seguir mostra como usar ListDatastores.

AWS CLI com 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
}
```

- Para API obter detalhes, consulte [ListDatastores](#) na Referência de AWS CLI Comandos.

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da AWS](#).

IAM exemplos de uso AWS CLI com o script Bash

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o script AWS Command Line Interface with Bash with IAM.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Os cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)
- [Cenários](#)

Ações

AttachRolePolicy

O código de exemplo a seguir mostra como usar AttachRolePolicy.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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 getopts "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_arn="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
```

```

aws_cli_error_log $error_code
errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
return 1
fi

echo "$response"

return 0
}

```

- Para API obter detalhes, consulte [AttachRolePolicy](#) na Referência de AWS CLI Comandos.

CreateAccessKey

O código de exemplo a seguir mostra como usar CreateAccessKey.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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 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
}
```

- Para API obter detalhes, consulte [CreateAccessKey](#) na Referência de AWS CLI Comandos.

CreatePolicy

O código de exemplo a seguir mostra como usar CreatePolicy.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
        esac
    done
}
```

```
    ;;
    \(?)
    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"
}
```

- Para API obter detalhes, consulte [CreatePolicy](#) na Referência de AWS CLI Comandos.

CreateRole

O código de exemplo a seguir mostra como usar CreateRole.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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 getopts "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_document="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
    --query Role.Arn)

local error_code=${?}
```

```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

```

- Para API obter detalhes, consulte [CreateRole](#) na Referência de AWS CLI Comandos.

CreateUser

O código de exemplo a seguir mostra como usar CreateUser.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [CreateUser](#) na Referência de AWS CLI Comandos.

DeleteAccessKey

O código de exemplo a seguir mostra como usar DeleteAccessKey.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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 getopts "u:k:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  Username:  $user_name"
iecho "  Access key: $access_key"
iecho ""

response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")
```

```

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
fi

iecho "delete-access-key response:$response"
iecho

return 0
}

```

- Para API obter detalhes, consulte [DeleteAccessKey](#) na Referência de AWS CLI Comandos.

DeletePolicy

O código de exemplo a seguir mostra como usar DeletePolicy.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

```
        ;;
    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
}
```

- Para API obter detalhes, consulte [DeletePolicy](#) na Referência de AWS CLI Comandos.

DeleteRole

O código de exemplo a seguir mostra como usar DeleteRole.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

- Para API obter detalhes, consulte [DeleteRole](#) na Referência de AWS CLI Comandos.

DeleteUser

O código de exemplo a seguir mostra como usar DeleteUser.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```



```
    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
}
```

- Para API obter detalhes, consulte [DeleteUser](#) na Referência de AWS CLI Comandos.

DetachRolePolicy

O código de exemplo a seguir mostra como usar DetachRolePolicy.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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 getopts "n:p:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    p) policy_arn="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_arn" ]]; then
  errecho "ERROR: You must provide a policy ARN with the -p parameter."
  usage
  return 1
fi

response=$(aws iam detach-role-policy \
  --role-name "$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
}

```

- Para API obter detalhes, consulte [DetachRolePolicy](#) na Referência de AWS CLI Comandos.

GetUser

O código de exemplo a seguir mostra como usar GetUser.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [GetUser](#) na Referência de AWS CLI Comandos.

ListAccessKeys

O código de exemplo a seguir mostra como usar ListAccessKeys.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [ListAccessKeys](#) na Referência de AWS CLI Comandos.

ListUsers

O código de exemplo a seguir mostra como usar ListUsers.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [ListUsers](#) na Referência de AWS CLI Comandos.

Cenários

Criar um usuário e assumir uma função

O exemplo de código a seguir mostra como criar um usuário e assumir um perfil.

⚠ Warning

Para evitar riscos de segurança, não use IAM usuários para autenticação ao desenvolver software específico ou trabalhar com dados reais. Em vez disso, use federação com um provedor de identidade, como [AWS IAM Identity Center](#).

- Crie um usuário sem permissões.
- Crie uma função que conceda permissão para listar os buckets do Amazon S3 para a conta.
- Adicione uma política para permitir que o usuário assuma a função.
- Assuma o perfil e liste buckets do S3 usando credenciais temporárias, depois limpe os recursos.

AWS CLI com script Bash

i Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
```

```
}
```

As IAM funções usadas nesse cenário.

```
#####  
# 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 getopts "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_document="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
fi

response=$(aws iam create-role \
    --role-name "$role_name" \
    --assume-role-policy-document "$policy_document" \
    --output text \
```

```

    --query Role.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# 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"
}
```

```
#####
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a role.
#
# Parameters:
#   -n role_name -- The name of the IAM role.
#   -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#   0 - If successful.
#   1 - If it fails.
#####
function iam_attach_role_policy() {
    local role_name policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_attach_role_policy"
        echo "Attaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
        echo "  -n role_name  The name of the IAM role."
        echo "  -p policy_ARN -- The IAM policy document ARN."
        echo ""
    }

    # Retrieve the calling parameters.
    while getopt "n:p:h" option; do
        case "${option}" in
            n) role_name="${OPTARG}" ;;
            p) policy_arn="${OPTARG}" ;;
            h)
                usage
                return 0
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
                ;;
        esac
    done
    export OPTIND=1
}
```



```

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam attach-role-policy \
    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports attach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a role.
#
# Parameters:
#     -n role_name -- The name of the IAM role.
#     -p policy_ARN -- The IAM policy document ARN..
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_detach_role_policy() {

```

```
local role_name policy_arn response
local option OPTARG # Required to use getopt command in a function.

# bashsupport disable=BP5008
function usage() {
    echo "function iam_detach_role_policy"
    echo "Detaches an AWS Identity and Access Management (IAM) policy to an IAM
role."
    echo "  -n role_name    The name of the IAM role."
    echo "  -p policy_ARN -- The IAM policy document ARN."
    echo ""
}

# Retrieve the calling parameters.
while getopt "n:p:h" option; do
    case "${option}" in
        n) role_name="${OPTARG}" ;;
        p) policy_arn="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
fi

if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy ARN with the -p parameter."
    usage
    return 1
fi

response=$(aws iam detach-role-policy \
```

```

    --role-name "$role_name" \
    --policy-arn "$policy_arn")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports detach-role-policy operation failed.\n$response"
    return 1
fi

echo "$response"

return 0
}

#####
# function iam_delete_policy
#
# This function deletes an IAM policy.
#
# Parameters:
#     -n policy_arn -- The name of the IAM policy arn.
#
# Returns:
#     0 - If successful.
#     1 - If it fails.
#####
function iam_delete_policy() {
    local policy_arn response
    local option OPTARG # Required to use getopt command in a function.

    # bashsupport disable=BP5008
    function usage() {
        echo "function iam_delete_policy"
        echo "Deletes an WS Identity and Access Management (IAM) policy"
        echo "  -n policy_arn -- The name of the IAM policy arn."
        echo ""
    }

    # Retrieve the calling parameters.
    while 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
}

#####
# 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 getopts "u:k:h" option; do
    case "${option}" in
        u) user_name="${OPTARG}" ;;
        k) access_key="${OPTARG}" ;;
        h)
            usage
            return 0
            ;;
        \?)
            echo "Invalid parameter"
            usage
            return 1
            ;;
    esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
fi

if [[ -z "$access_key" ]]; then
    errecho "ERROR: You must provide an access key with the -k parameter."
    usage
    return 1
fi

iecho "Parameters:\n"
iecho "  Username:  $user_name"
iecho "  Access key:  $access_key"
iecho ""

response=$(aws iam delete-access-key \
    --user-name "$user_name" \
    --access-key-id "$access_key")

local error_code=${?}
```

```

if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-access-key operation failed.\n$response"
    return 1
fi

iecho "delete-access-key response:$response"
iecho

return 0
}

#####
# 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
}
```

- Para API obter detalhes, consulte os tópicos a seguir na Referência de AWS CLI Comandos.
 - [AttachRolePolicy](#)
 - [CreateAccessKey](#)
 - [CreatePolicy](#)
 - [CreateRole](#)
 - [CreateUser](#)
 - [DeleteAccessKey](#)
 - [DeletePolicy](#)
 - [DeleteRole](#)
 - [DeleteUser](#)
 - [DeleteUserPolicy](#)
 - [DetachRolePolicy](#)
 - [PutUserPolicy](#)

Exemplos do Amazon S3 usando o script AWS CLI Bash

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o script AWS Command Line Interface with Bash com o Amazon S3.

As noções básicas são exemplos de código que mostram como realizar as operações essenciais em um serviço.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Conceitos básicos](#)
- [Ações](#)

Conceitos básicos

Aprender os conceitos básicos

O exemplo de código a seguir mostra como:

- Criar um bucket e fazer upload de um arquivo para ele.
- Baixar um objeto de um bucket.
- Copiar um objeto em uma subpasta em um bucket.
- Listar os objetos em um bucket.
- Excluir os objetos do bucket e o bucket.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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

local bucket_name
bucket_name=$(generate_random_name "doc-example-bucket")

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
}

```

As funções do Amazon S3 usadas nesse cenário.

```

#####
# 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
}

#####
# 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
}

```


- Para API obter detalhes, consulte os tópicos a seguir na Referência de AWS CLI Comandos.
 - [CopyObject](#)
 - [CreateBucket](#)
 - [DeleteBucket](#)
 - [DeleteObjects](#)
 - [GetObject](#)
 - [ListObjectsV2](#)
 - [PutObject](#)

Ações

CopyObject

O código de exemplo a seguir mostra como usar CopyObject.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [CopyObject](#) na Referência de AWS CLI Comandos.

CreateBucket

O código de exemplo a seguir mostra como usar CreateBucket.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [CreateBucket](#) na Referência de AWS CLI Comandos.

DeleteBucket

O código de exemplo a seguir mostra como usar DeleteBucket.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

- Para API obter detalhes, consulte [DeleteBucket](#) na Referência de AWS CLI Comandos.

DeleteObject

O código de exemplo a seguir mostra como usar DeleteObject.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [DeleteObject](#) na Referência de AWS CLI Comandos.

DeleteObjects

O código de exemplo a seguir mostra como usar DeleteObjects.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}


```

- Para API obter detalhes, consulte [DeleteObjects](#) na Referência de AWS CLI Comandos.

GetObject

O código de exemplo a seguir mostra como usar GetObject.

AWS CLI com script Bash

 Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}
```

- Para API obter detalhes, consulte [GetObject](#) na Referência de AWS CLI Comandos.

HeadBucket

O código de exemplo a seguir mostra como usar HeadBucket.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

- Para API obter detalhes, consulte [HeadBucket](#) na Referência de AWS CLI Comandos.

ListObjectsV2

O código de exemplo a seguir mostra como usar ListObjectsV2.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

- Para API obter detalhes, consulte [ListObjectsV2](#) na Referência de AWS CLI Comandos.

PutObject

O código de exemplo a seguir mostra como usar PutObject.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
}

```

- Para API obter detalhes, consulte [PutObject](#) na Referência de AWS CLI Comandos.

AWS STS exemplos de uso AWS CLI com o script Bash

Os exemplos de código a seguir mostram como realizar ações e implementar cenários comuns usando o script AWS Command Line Interface with Bash with AWS STS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cada exemplo inclui um link para o código-fonte completo, onde você pode encontrar instruções sobre como configurar e executar o código no contexto.

Tópicos

- [Ações](#)

Ações

AssumeRole

O código de exemplo a seguir mostra como usar AssumeRole.

AWS CLI com script Bash

Note

Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no [Repositório de exemplos de código da 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
        ;;
    \?)
        echo "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
}
```

- Para API obter detalhes, consulte [AssumeRole](#) na Referência de AWS CLI Comandos.

Segurança na AWS Command Line Interface

A segurança na nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — a AWS é responsável pela proteção da infraestrutura que executa serviços AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos [AWS Programas de Conformidade](#). Para saber mais sobre os programas de conformidade que se aplicam ao AWS Command Line Interface, consulte [AWS Serviços da em escopo por programa de conformidade](#).
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a AWS Command Line Interface (AWS CLI). Os tópicos a seguir mostram como configurar a AWS CLI para atender aos seus objetivos de segurança e conformidade. Você também aprende como usar a AWS CLI para ajudar a monitorar e a proteger os recursos da AWS.

Tópicos

- [Proteção de dados no AWS CLI](#)
- [Identity and Access Management](#)
- [Validação de conformidade para este AWS produto ou serviço](#)
- [Resiliência para este AWS produto ou serviço](#)
- [Segurança da infraestrutura para este AWS produto ou serviço](#)
- [Aplique uma versão mínima do TLS](#)

Proteção de dados no AWS CLI

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS Command Line Interface. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS CLI ou Serviços da AWS usa o console, API, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se

Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

Criptografia de dados

Um atributo fundamental de qualquer serviço seguro é que as informações sejam criptografadas quando não estão sendo usadas ativamente.

Criptografia em repouso

O próprio AWS CLI não armazena nenhum dado do cliente além das credenciais necessárias para interagir com os AWS serviços em nome do usuário.

Se você usar o AWS CLI para invocar um AWS serviço que transmite dados do cliente para seu computador local para armazenamento, consulte o capítulo Segurança e Conformidade no Guia do Usuário desse serviço para obter informações sobre como esses dados são armazenados, protegidos e criptografados.

Criptografia em trânsito

Por padrão, todos os dados transmitidos do computador cliente que executa os endpoints AWS de serviço AWS CLI e são criptografados enviando tudo por meio de uma TLS conexão HTTPS /.

Você não precisa fazer nada para habilitar o uso de HTTPS/TLS. Ele está sempre ativado, a menos que você o desative explicitamente para um comando individual usando a opção de linha de comando `--no-verify-ssl`.

Identity and Access Management

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS os recursos. IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)

- [Como Serviços da AWS trabalhar com IAM](#)
- [Solução de problemas AWS de identidade e acesso](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz AWS.

Usuário do serviço — Se você Serviços da AWS costuma fazer seu trabalho, seu administrador fornece as credenciais e as permissões de que você precisa. À medida que você usa mais AWS recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso no AWS, consulte [Solução de problemas AWS de identidade e acesso](#) o guia do usuário do AWS service (Serviço da AWS) que você está usando.

Administrador de serviços — Se você é responsável pelos AWS recursos da sua empresa, provavelmente tem acesso total AWS a. É seu trabalho determinar quais AWS recursos e recursos seus usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM com AWS, consulte o guia do usuário do AWS service (Serviço da AWS) que você está usando.

IAM administrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso AWS. Para ver exemplos de políticas AWS baseadas em identidade que você pode usar em IAM, consulte o guia do usuário do AWS service (Serviço da AWS) que você está usando.

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu

administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [Assinar AWS API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia AWS IAM Identity Center do usuário e [Uso da autenticação multifator \(MFA\) AWS no Guia do IAMusuário](#).

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um IAM usuário \(em vez de uma função\)](#) no Guia do IAM usuário.

IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Usando IAM funções](#) no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa

identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter informações sobre funções para federação, consulte [Criação de uma função para um provedor de identidade terceirizado](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .

- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a um serviço.
- Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- Função de serviço — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS) O serviço pode presumir a função de executar

uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte [Quando criar uma IAM função \(em vez de um usuário\)](#) no Guia do IAM usuário.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolha entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões** — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para IAM entidades](#) no Guia IAM do usuário.
- **Políticas de controle de serviço (SCPs)** — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia IAM do usuário.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

Como Serviços da AWS trabalhar com IAM

Para ter uma visão geral de como Serviços da AWS funciona com a maioria dos IAM recursos, consulte [AWS os serviços que funcionam com IAM](#) no Guia do IAM usuário.

Para saber como usar um AWS service (Serviço da AWS) com específico IAM, consulte a seção de segurança do Guia do usuário do serviço relevante.

Solução de problemas AWS de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS e IAM

Tópicos

- [Não estou autorizado a realizar uma ação em AWS](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS recursos](#)

Não estou autorizado a realizar uma ação em AWS

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O exemplo de erro a seguir ocorre quando o mateojackson IAM usuário tenta usar o console para ver detalhes sobre um *my-example-widget* recurso fictício, mas não tem as permissões fictícias `aws:GetWidget`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário mateojackson deve ser atualizada para permitir o acesso ao recurso *my-example-widget* usando a ação `aws:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar uma ação no AWS. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus AWS recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é AWS compatível com esses recursos, consulte [Como Serviços da AWS trabalhar com IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outro Conta da AWS de sua propriedade](#) no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.

- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

Validação de conformidade para este AWS produto ou serviço

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para HIPAA segurança e conformidade na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

Note

Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a [Referência de serviços HIPAA elegíveis](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização ()). ISO

- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Esse AWS produto ou serviço segue o [modelo de responsabilidade compartilhada](#) por meio dos serviços específicos da Amazon Web Services (AWS) que ele suporta. Para AWS obter informações sobre segurança do [AWS serviço](#), consulte a [página de documentação de segurança](#) do serviço e os [AWS serviços que estão no escopo dos esforços de AWS conformidade do programa de conformidade](#).

Resiliência para este AWS produto ou serviço

A infraestrutura AWS global é construída em torno Regiões da AWS de zonas de disponibilidade.

Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância.

Com as zonas de disponibilidade, é possível projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Esse AWS produto ou serviço segue o [modelo de responsabilidade compartilhada](#) por meio dos serviços específicos da Amazon Web Services (AWS) que ele suporta. Para AWS obter informações sobre segurança do [AWS serviço, consulte a página de documentação de segurança](#) do serviço e os [AWS serviços que estão no escopo dos esforços de AWS conformidade do programa de conformidade](#).

Segurança da infraestrutura para este AWS produto ou serviço

Esse AWS produto ou serviço usa serviços gerenciados e, portanto, é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa API chamadas AWS publicadas para acessar este AWS Produto ou Serviço pela rede. Os clientes devem oferecer suporte para:

- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Esse AWS produto ou serviço segue o [modelo de responsabilidade compartilhada](#) por meio dos serviços específicos da Amazon Web Services (AWS) que ele suporta. Para AWS obter informações sobre segurança do [AWS serviço, consulte a página de documentação de segurança](#) do serviço e os [AWS serviços que estão no escopo dos esforços de AWS conformidade do programa de conformidade](#).

Aplice uma versão mínima do TLS

Ao usar o AWS Command Line Interface (AWS CLI), o protocolo Transport Layer Security (TLS) desempenha um papel crucial na proteção da comunicação entre AWS CLI e. Serviços da AWS Para aumentar a segurança ao se comunicar com os AWS serviços, você deve usar a versão TLS 1.2 ou posterior.

O AWS CLI and AWS service (Serviço da AWS) pode trocar dados com segurança, com o TLS protocolo fornecendo criptografia, autenticação e integridade dos dados. Ao aproveitar o TLS protocolo, ele AWS CLI garante que suas interações Serviços da AWS sejam protegidas contra acesso não autorizado e violações de dados, aumentando a segurança geral do seu ecossistema. AWS

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS Command Line Interface. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Serviços da AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança do Serviços da AWS que você usa. Para obter mais informações sobre proteção de dados, consulte [the section called “Proteção de dados”](#).

Para garantir que a AWS CLI versão 1 não use nenhuma TLS versão anterior à TLS 1.2, talvez seja necessário recompilar o Open SSL para impor esse mínimo e, em seguida, recompilar o Python para usar o Open recém-criado. SSL

Tópicos

- [Determinar os protocolos atualmente compatíveis](#)
- [Compile Open SSL e Python](#)

Determinar os protocolos atualmente compatíveis

Primeiro, crie um certificado autoassinado para usar no servidor de teste e no SDK Python usando o Open. SSL

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Em seguida, ative um servidor de teste usando o OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

Em uma nova janela de terminal, crie um ambiente virtual e instale o SDK para Python.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Crie um novo script Python chamado `check.py` que use a biblioteca subjacente SDK HTTP do.

```
$ 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'))
```

Execute o novo script.

```
$ python check.py
```

Ele exibe detalhes sobre a conexão feita. Procure “Protocol :” na saída. Se a saída for “TLSv1.2” ou posterior, o SDK padrão será v1.2 ou posterior. TLS Se for uma versão anterior, você precisará recompilar o Open SSL e recompilar o Python.

No entanto, mesmo que sua instalação do Python tenha como padrão a versão TLS 1.2 ou posterior, ainda é possível que o Python renegocie para uma versão anterior à TLS v1.2 se o servidor não suportar a versão 1.2 ou posterior. TLS Para verificar se o Python não renegocia automaticamente para versões anteriores, reinicie o servidor de teste com o seguinte.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Se você estiver usando uma versão anterior do OpenSSL, talvez não tenha a `-no_tls1_3` bandeira disponível. Se for esse o caso, remova o sinalizador porque a versão do Open SSL que você está usando não é compatível com a TLS v1.3. Execute novamente o script Python.

```
$ python check.py
```

Se sua instalação do Python corretamente não renegociar para versões anteriores à TLS 1.2, você deverá receber um erro. 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 você conseguir fazer uma conexão, precisará recompilar o Open e o SSL Python para desativar a negociação de protocolos anteriores à v1.2. TLS

Compile Open SSL e Python

Para garantir que SDK ou AWS CLI não negocie nada anterior à TLS 1.2, você precisa recompilar o Open e SSL o Python. Para fazer isso, copie o seguinte conteúdo para criar um script e executá-lo.

```
#!/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
```

Isso compila uma versão do Python que tem um SSL Open vinculado estaticamente que não negocia automaticamente nada anterior à 1.2. TLS Isso também instala o Open SSL no /opt/openssl-with-min-tls1_2 diretório e instala o Python no diretório. /opt/python-with-min-tls1_2 Depois de executar esse script, verifique a instalação da nova versão do Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Isso deve imprimir o seguinte.

```
$ Python 3.8.1
```

Para confirmar que essa nova versão do Python não negocia uma versão anterior à TLS 1.2, execute novamente as etapas de uso da versão recém-instalada [Determinar os protocolos atualmente compatíveis](#) do Python (ou seja,). /opt/python-with-min-tls1_2/bin/python3

Solucionar problemas de erros AWS CLI

Esta seção aborda erros comuns e os passos para a solução de problemas a serem seguidas para resolver o problema. Sugerimos seguir a [Solução geral de problemas](#) primeiro.

Sumário

- [Solução geral de problemas para tentar primeiro](#)
 - [Verifique a formatação AWS CLI do seu comando](#)
 - [Verifique o que Região da AWS seu AWS CLI comando está usando](#)
 - [Confirme se você está executando uma versão recente da AWS CLI](#)
 - [Como usar a opção --debug](#)
 - [Ative e revise os registros do histórico de AWS CLI comandos](#)
 - [Confirme se o seu AWS CLI está configurado](#)
- [Erros de comando não encontrado](#)
- [O comando "aws --version" retorna uma versão diferente da que você instalou](#)
- [O comando "aws --version" retorna uma versão após a desinstalação do AWS CLI](#)
- [O comando AWS CLI processado com um nome de parâmetro incompleto](#)
- [Erros de acesso negado](#)
- [Credenciais inválidas e erros de chave](#)
- [Assinatura não corresponde aos erros](#)
- [Erros de nenhum console do Windows encontrado](#)
- [SSLerros de certificado](#)
- [Erros inválidos JSON](#)
- [Recursos adicionais](#)

Solução geral de problemas para tentar primeiro

Se você receber um erro ou encontrar um problema com o AWS CLI, sugerimos as dicas gerais a seguir para ajudá-lo a solucionar o problema.

[Voltar ao início](#)

Verifique a formatação AWS CLI do seu comando

Se você receber um erro indicando que um comando não existe ou que ele não reconhece um parâmetro que a documentação indica estar disponível, é provável que o comando esteja formatado incorretamente. Sugerimos verificar o seguinte:

- Verifique se há erros de ortografia e de formatação no comando.
- Confirme se todos os [escapes e citações apropriados para o seu terminal](#) estão corretos em seu comando.
- Gere um [esqueleto de AWS CLI](#) para confirmar a estrutura do comando.
- Para obter informações sobre [JSONvaloresJSON](#), consulte [a solução de problemas](#) adicional. Se você estiver tendo problemas com a JSON formatação do processamento do terminal, sugerimos ignorar as regras de cotação do terminal usando [Blobs para passar JSON dados diretamente para o. AWS CLI](#)

Para obter mais informações sobre como um comando específico deve ser estruturado, consulte o guia de [AWS CLI referência da AWS CLI versão 2 do guia](#) .

[Voltar ao início](#)

Verifique o que Região da AWS seu AWS CLI comando está usando

Note

Você deve especificar um Região da AWS ao usar o AWS CLI, explicitamente ou definindo uma região padrão. Para obter uma lista de tudo o Regiões da AWS que você pode especificar, consulte [AWS Regiões e endpoints](#) no Referência geral da Amazon Web Services. Os Região da AWS designadores usados pelo AWS CLI são os mesmos nomes que você vê nos endpoints AWS Management Console URLs de serviço.

Erros ou resultados inesperados podem ocorrer se um AWS service (Serviço da AWS) não estiver disponível para o que você especificou Região da AWS ou se seus recursos estiverem localizados em um local diferente Região da AWS. Em ordem de precedência, o Região da AWS é definido das seguintes maneiras:

- A opção da linha de comando `--region`.

- A variável de ambiente [AWS_DEFAULT_REGION](#).
- A configuração [region](#) do perfil.

Confirme se você está usando o correto Região da AWS para seus recursos.

[Voltar ao início](#)

Confirme se você está executando uma versão recente da AWS CLI

Se você receber um erro indicando que um comando não existe ou que ele não reconhece um parâmetro que o guia de [AWS CLI referência da AWS CLI versão 2 do guia](#) diz estar disponível, primeiro confirme se o comando está formatado corretamente. Se a formatação estiver correta, recomendamos atualizar para a versão mais recente da AWS CLI. Versões atualizadas do AWS CLI são lançadas quase todos os dias úteis. Novos AWS serviços, recursos e parâmetros são introduzidos nessas novas versões do AWS CLI. A única maneira de obter acesso a esses novos serviços, recursos ou parâmetros é atualizar para uma versão que foi lançada depois que esse elemento foi apresentado pela primeira vez.

A forma como você atualiza sua versão do AWS CLI depende de como você a instalou originalmente, conforme descrito em [Instalar a AWS CLI](#).

Se você tiver usado um dos instaladores empacotados, talvez precise remover a instalação existente antes de baixar e instalar a versão mais recente para seu sistema operacional.

[Voltar ao início](#)

Como usar a opção **--debug**

Quando ele AWS CLI relata um erro que você não entende imediatamente ou produz resultados inesperados, você pode obter mais detalhes sobre o erro executando o comando novamente com a **--debug** opção. Com esta opção, a AWS CLI gera detalhes sobre cada passo necessário para processar seu comando. Os detalhes na saída podem ajudar a determinar quando o erro ocorreu e fornecem dicas sobre o que o acionou o erro.

É possível enviar a saída para um arquivo de texto para análise posterior ou enviá-lo para AWS Support quando solicitado.

Quando você inclui a opção **--debug**, os detalhes incluem:

- Procurar credenciais

- Analisar os parâmetros fornecidos
- Construindo a solicitação enviada aos servidores AWS
- O conteúdo da solicitação enviada para AWS
- O conteúdo da resposta não formatada
- A saída formatada

Veja a seguir um exemplo de um comando executado com e sem a opção `--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 - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_streaming_output_arg at
0x7fdf17316510>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_cli_input_json at
0x7fdf17da9d90>
2019-08-12 12:36:18,322 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function unify_paging_params at
0x7fdf17328048>
2019-08-12 12:36:18,326 - MainThread - botocore.loaders - DEBUG - Loading JSON
file: /home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/iam/2010-05-08/
paginator-1.json
2019-08-12 12:36:18,326 - MainThread - awscli.customizations.paginate - DEBUG -
Modifying paging parameters for operation: ListGroups
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event building-
argument-table.iam.list-groups: calling handler <function add_generate_skeleton at
0x7fdf1737eae8>
2019-08-12 12:36:18,326 - MainThread - botocore.hooks - DEBUG - Event
before-building-argument-table-parser.iam.list-groups: calling handler
<bound method OverrideRequiredArgsArgument.override_required_args of
<awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event
before-building-argument-table-parser.iam.list-groups: calling handler
<bound method GenerateCliSkeletonArgument.override_required_args of
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>>
2019-08-12 12:36:18,327 - MainThread - botocore.hooks - DEBUG - Event operation-
args-parsed.iam.list-groups: calling handler functools.partial(<function
check_should_enable_pagination at 0x7fdf17328158>, ['marker', 'max-items'], {'max-
items': <awscli.arguments.CLIArgument object at 0x7fdf171b09b0>}, OrderedDict([('path-
prefix', <awscli.arguments.CLIArgument object at 0x7fdf171ac780>), ('marker',
<awscli.arguments.CLIArgument object at 0x7fdf171b09e8>), ('max-items',
<awscli.customizations.paginate.PageArgument object at 0x7fdf171c58d0>), ('cli-
input-json', <awscli.customizations.cliinputjson.CliInputJSONArgument object at
0x7fdf171b0a58>), ('starting-token', <awscli.customizations.paginate.PageArgument
```

```
object at 0x7fdf171b0a20>), ('page-size', <awscli.customizations.paginate.PageArgument
object at 0x7fdf171c5828>), ('generate-cli-skeleton',
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>]]))
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.path-prefix: calling handler <awscli.paramfile.URIArgumentHandler
object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.marker: calling handler <awscli.paramfile.URIArgumentHandler object
at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.max-items: calling handler <awscli.paramfile.URIArgumentHandler
object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
Event load-cli-arg.iam.list-groups.cli-input-json: calling handler
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG -
Event load-cli-arg.iam.list-groups.starting-token: calling handler
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event load-cli-
arg.iam.list-groups.page-size: calling handler <awscli.paramfile.URIArgumentHandler
object at 0x7fdf1725c978>
2019-08-12 12:36:18,328 - MainThread - botocore.hooks - DEBUG - Event
load-cli-arg.iam.list-groups.generate-cli-skeleton: calling handler
<awscli.paramfile.URIArgumentHandler object at 0x7fdf1725c978>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG
- Event calling-command.iam.list-groups: calling handler
<bound method CliInputJSONArgument.add_to_call_parameters of
<awscli.customizations.cliinputjson.CliInputJSONArgument object at 0x7fdf171b0a58>>
2019-08-12 12:36:18,329 - MainThread - botocore.hooks - DEBUG -
Event calling-command.iam.list-groups: calling handler <bound
method GenerateCliSkeletonArgument.generate_json_skeleton of
<awscli.customizations.generatecliskeleton.GenerateCliSkeletonArgument object at
0x7fdf171c5978>>
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
credentials via: assume-role
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
credentials via: assume-role-with-web-identity
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - DEBUG - Looking for
credentials via: shared-credentials-file
2019-08-12 12:36:18,329 - MainThread - botocore.credentials - INFO - Found credentials
in shared credentials file: ~/.aws/credentials
2019-08-12 12:36:18,330 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/endpoints.json
```

```
2019-08-12 12:36:18,334 - MainThread - botocore.hooks - DEBUG - Event choose-service-
name: calling handler <function handle_service_name_alias at 0x7fdf1898eb70>
2019-08-12 12:36:18,337 - MainThread - botocore.hooks - DEBUG - Event creating-client-
class.iam: calling handler <function add_generate_presigned_url at 0x7fdf18a028c8>
2019-08-12 12:36:18,337 - MainThread - botocore.regions - DEBUG - Using partition
endpoint for iam, us-west-2: aws-global
2019-08-12 12:36:18,337 - MainThread - botocore.args - DEBUG - The s3 config key is not
a dictionary type, ignoring its value of: None
2019-08-12 12:36:18,340 - MainThread - botocore.endpoint - DEBUG - Setting iam timeout
as (60, 60)
2019-08-12 12:36:18,341 - MainThread - botocore.loaders - DEBUG - Loading JSON file: /
home/ec2-user/venv/lib/python3.7/site-packages/botocore/data/_retry.json
2019-08-12 12:36:18,341 - MainThread - botocore.client - DEBUG - Registering retry
handlers for service: iam
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
parameter-build.iam.ListGroups: calling handler <function generate_idempotent_uuid at
0x7fdf189b10d0>
2019-08-12 12:36:18,342 - MainThread - botocore.hooks - DEBUG - Event before-
call.iam.ListGroups: calling handler <function inject_api_version_header_if_needed at
0x7fdf189b2a60>
2019-08-12 12:36:18,343 - MainThread - botocore.endpoint - DEBUG - Making
request for OperationModel(name=ListGroups) with params: {'url_path': '/',
'query_string': '', 'method': 'POST', 'headers': {'Content-Type': 'application/x-
www-form-urlencoded; charset=utf-8', 'User-Agent': 'aws-cli/1.16.215 Python/3.7.3
Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205'}, 'body': {'Action':
'ListGroups', 'Version': '2010-05-08'}, 'url': 'https://iam.amazonaws.com/',
'context': {'client_region': 'aws-global', 'client_config': <botocore.config.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
<botocore.signers.RequestSigner object at 0x7fdf16e9a470>>
2019-08-12 12:36:18,343 - MainThread - botocore.hooks - DEBUG - Event choose-
signer.iam.ListGroups: calling handler <function set_operation_specific_signer at
0x7fdf18996f28>
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - Calculating signature
using v4 auth.
2019-08-12 12:36:18,343 - MainThread - botocore.auth - DEBUG - CanonicalRequest:
POST
/

content-type:application/x-www-form-urlencoded; charset=utf-8
host:iam.amazonaws.com
x-amz-date:20190812T193618Z
```

```

content-type;host;x-amz-date
5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEEca69080e1e1f
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - StringToSign:
AWS4-HMAC-SHA256
20190812T193618Z
20190812/us-east-1/iam/aws4_request
ab7e367eEXAMPLE2769f178ea509978cf8bfa054874b3EXAMPLE8d043fab6cc9
2019-08-12 12:36:18,344 - MainThread - botocore.auth - DEBUG - Signature:
d85a0EXAMPLEeb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
2019-08-12 12:36:18,344 - MainThread - botocore.endpoint - DEBUG - Sending
  http request: <AWSPreparedRequest stream_output=False, method=POST,
  url=https://iam.amazonaws.com/, headers={'Content-Type': b'application/
x-www-form-urlencoded; charset=utf-8', 'User-Agent': b'aws-cli/1.16.215
Python/3.7.3 Linux/4.14.133-113.105.amzn2.x86_64 botocore/1.12.205',
'X-Amz-Date': b'20190812T193618Z', 'Authorization': b'AWS4-HMAC-SHA256
Credential=AKIA01234567890EXAMPLE-east-1/iam/aws4_request, SignedHeaders=content-
type;host;x-amz-date, Signature=d85a07692aceb401EXAMPLEa1b18ad1ddf58a1a3ce7EXAMPLE',
'Content-Length': '36'}>
2019-08-12 12:36:18,344 - MainThread - urllib3.util.retry - DEBUG - Converted retries
value: False -> Retry(total=False, connect=None, read=None, redirect=0, status=None)
2019-08-12 12:36:18,344 - MainThread - urllib3.connectionpool - DEBUG - Starting new
HTTPS connection (1): iam.amazonaws.com:443
2019-08-12 12:36:18,664 - MainThread - urllib3.connectionpool - DEBUG - https://
iam.amazonaws.com:443 "POST / HTTP/1.1" 200 570
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response headers:
{'x-amzn-RequestId': '74c11606-bd38-11e9-9c82-559da0adb349', 'Content-Type': 'text/
xml', 'Content-Length': '570', 'Date': 'Mon, 12 Aug 2019 19:36:18 GMT'}
2019-08-12 12:36:18,664 - MainThread - botocore.parsers - DEBUG - Response body:
b'<ListGroupResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">\n
  <ListGroupResult>\n    <IsTruncated>>false</IsTruncated>\n    <Groups>\n
    <member>\n      <Path>/</Path>\n      <GroupName>MyTestGroup</GroupName>
\n      <Arn>arn:aws:iam::123456789012:group/MyTestGroup</Arn>\n
    <GroupId>AGPA1234567890EXAMPLE</GroupId>\n      <CreateDate>2019-08-12T19:34:04Z</
CreateDate>\n    </member>\n    </Groups>\n  </ListGroupResult>\n
  <ResponseMetadata>\n    <RequestId>74c11606-bd38-11e9-9c82-559da0adb349</RequestId>\n
  </ResponseMetadata>\n</ListGroupResponse>\n'
2019-08-12 12:36:18,665 - MainThread - botocore.hooks - DEBUG - Event needs-
retry.iam.ListGroups: calling handler <botocore.retryhandler.RetryHandler object at
0x7fdf16e9a780>
2019-08-12 12:36:18,665 - MainThread - botocore.retryhandler - DEBUG - No retry needed.
2019-08-12 12:36:18,665 - MainThread - botocore.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"
}
]
```

[Voltar ao início](#)

Ative e revise os registros do histórico de AWS CLI comandos

Você pode ativar os registros do histórico de AWS CLI comandos usando a configuração [cli_history](#) do arquivo. Depois de ativar essa configuração, ele AWS CLI registra o histórico dos aws comandos.

Você pode listar seu histórico usando o comando `aws history list` e usar os `command_ids` resultantes no comando `aws history show` para obter detalhes. Para obter mais informações, consulte [aws history](#) no guia de referência da AWS CLI .

Quando você inclui a opção `--debug`, os detalhes incluem:

- API chamadas feitas para botocore
- Códigos de status
- HTTP respostas
- Cabeçalhos
- Códigos de retorno

Você pode usar essas informações para confirmar que os dados dos parâmetros e as API chamadas estão se comportando da maneira esperada e, em seguida, deduzir em qual etapa do processo seu comando está falhando.

[Voltar ao início](#)

Confirme se o seu AWS CLI está configurado

Vários erros podem ocorrer se seus `credentials` arquivos `config` e ou seu IAM usuário ou função não estiverem configurados corretamente. Para obter mais informações sobre como resolver

erros com `credentials` arquivos `config` e ou com seu IAM usuário ou funções, consulte [the section called “Erros de acesso negado”](#) e [the section called “Credenciais inválidas e erros de chave”](#)

[Voltar ao início](#)

Erros de comando não encontrado

Esse erro significa que o sistema operacional não consegue encontrar o AWS CLI comando. A instalação pode estar incompleta ou exigir atualização.

Possível causa: você está tentando usar um AWS CLI recurso mais recente do que a versão instalada ou tem uma formatação incorreta

Exemplo de texto de erro:

```
$ 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
....
```

Vários erros podem ocorrer se o comando estiver formatado incorretamente ou se você estiver usando uma versão anterior ao lançamento do recurso. Para obter mais informações sobre a solução de erros em torno desses dois problemas, consulte [the section called “Verifique a formatação AWS CLI do seu comando”](#) e [the section called “Confirme se você está executando uma versão recente da AWS CLI”](#).

[Voltar ao início](#)

Causa possível: o terminal precisa ser reiniciado após a instalação

Exemplo de texto de erro:

```
$ aws --version
```

```
command not found: aws
```

Se o aws comando não puder ser encontrado após a primeira instalação ou atualização do AWS CLI, talvez seja necessário reiniciar o terminal para que ele reconheça PATH as atualizações.

[Voltar ao início](#)

Possível causa: AWS CLI Não foi totalmente instalado

Exemplo de texto de erro:

```
$ aws --version  
command not found: aws
```

Se o aws comando não puder ser encontrado após a primeira instalação ou atualização do AWS CLI, talvez ele não tenha sido totalmente instalado. Tente reinstalar seguindo os passos da sua plataforma em [Instalar a AWS CLI](#).

[Voltar ao início](#)

Possível causa: AWS CLI Não tem permissões (Linux)

Se o aws comando não puder ser encontrado após a primeira instalação ou atualização do AWS CLI no Linux, talvez ele não tenha execute permissões para a pasta em que foi instalado. Execute o seguinte comando com o PATH para sua AWS CLI instalação, para fornecer [chmod](#) permissões para AWS CLI:

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

[Voltar ao início](#)

Causa possível: o **PATH** do sistema operacional não foi atualizado durante a instalação.

Exemplo de texto de erro:

```
$ aws --version  
command not found: aws
```

Talvez seja necessário adicionar o executável aws à variável de ambiente PATH do sistema operacional. Para adicionar o AWS CLI ao seuPATH, use as instruções a seguir para o seu sistema operacional.

Linux and macOS

1. Encontre o script de perfil do shell no diretório de usuário. Se não tiver certeza de qual shell você tem, execute `echo $SHELL`.

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

- Bash – `.bash_profile`, `.profile`, ou `.bash_login`
 - Zsh – `.zshrc`
 - Tcsh – `.tcshrc`, `.cshrc`, ou `.login`
2. Adicione um comando de exportação ao script de perfil. O comando a seguir adiciona seu compartimento local à variável `PATH` atual.

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

3. Recarregue o perfil atualizado em sua sessão atual.

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

Windows

1. Em um prompt de comando do Windows, use o comando `where` com o parâmetro `/R` *path* para encontrar o local do arquivo `aws`. Os resultados retornam todas as pastas que contêm `aws`.

```
C:\> where /R c:\ aws  
c:\Program Files\Amazon\AWSCLIV2\aws.exe  
...
```

Por padrão, a AWS CLI versão 2 está localizada em:

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. Pressione a tecla Windows e digite **environment variables**.
3. Na lista de sugestões, escolha `Edit environment variables for your account` (Editar variáveis de ambiente para sua conta).

4. Escolha e PATH, em seguida, escolha Editar.
5. Adicione o caminho encontrado no campo Variable value (Valor da variável). Por exemplo, **C:\Program Files\Amazon\AWSCLIV2\aws.exe**.
6. Escolha OK duas vezes para aplicar as novas configurações.
7. Feche todos os prompts de comando em execução e abra novamente a janela do prompt de comando.

[Voltar ao início](#)

O comando “**aws --version**” retorna uma versão diferente da que você instalou

Seu terminal pode estar retornando um valor diferente PATH do AWS CLI que você espera.

Causa possível: o terminal precisa ser reiniciado após a instalação

Se o comando `aws` mostrar a versão errada, talvez seja necessário reiniciar o terminal para que ele reconheça todas as atualizações de PATH. Todos os terminais abertos precisam estar fechados, não apenas o terminal ativo.

[Voltar ao início](#)

Causa possível: o sistema precisa ser reiniciado após a instalação

Se o comando `aws` mostrar a versão errada e reiniciar o terminal não funcionar, talvez seja necessário reiniciar o sistema para que ele reconheça as atualizações de PATH.

[Voltar ao início](#)

Possível causa: você tem várias versões do AWS CLI

Se você atualizou AWS CLI e usou um método de instalação diferente da instalação preexistente, isso pode fazer com que várias versões sejam instaladas. Por exemplo, se no Linux ou no macOS você usou `pip` para a instalação atual, mas tentou atualizar usando o arquivo de instalação `.pkg`, isso poderá causar alguns conflitos, especialmente com o PATH apontando para a versão antiga.

Para resolver isso, [desinstale todas as versões da AWS CLI](#) e execute uma instalação limpa.

Depois de desinstalar todas as versões, siga as instruções apropriadas do seu sistema operacional para instalar a versão desejada da [AWS CLI versão 1](#) ou da [AWS CLI versão 2](#).

Note

Se isso acontecer depois de instalar a AWS CLI versão 2 com uma instalação preexistente da AWS CLI versão 1, siga as instruções de migração em [Instruções de instalação ao migrar da AWS CLI versão 1](#).

[Voltar ao início](#)

O comando "**aws --version**" retorna uma versão após a desinstalação do AWS CLI

Isso geralmente ocorre quando ainda há um AWS CLI instalado em algum lugar do seu sistema.

Causa possível: o terminal precisa ser reiniciado após a instalação

Se o comando `aws --version` mostrar a versão errada, talvez seja necessário reiniciar o terminal para que ele reconheça todas as atualizações.

[Voltar ao início](#)

Possível causa: você tem várias versões do AWS CLI no seu sistema ou não usou o mesmo método de desinstalação usado para instalar originalmente o AWS CLI

AWS CLI Talvez não seja desinstalado corretamente se você tiver desinstalado o AWS CLI usando um método diferente do usado para instalá-lo ou se tiver instalado várias versões. Por exemplo, se você usou `pip` para a instalação atual, deverá usar `pip` para desinstalá-la. Para resolver isso, desinstale AWS CLI usando o mesmo método usado para instalá-lo.

1. Siga as instruções apropriadas do seu sistema operacional e do método de instalação original para desinstalar a [AWS CLI versão 1](#) e a [AWS CLI versão 2](#).
2. Feche todos os terminais que você abriu.
3. Abra seu terminal preferido, insira o seguinte comando e confirme que nenhuma versão é retornada.

```
$ aws --version
```

```
command not found: aws
```

Se você ainda tiver uma versão listada na saída, é provável AWS CLI que ela tenha sido instalada usando um método diferente ou que haja várias versões. Se você não souber qual método instalou o AWS CLI, siga as instruções de cada método de desinstalação das [AWS CLI versões 1](#) e [AWS CLI 2](#) apropriadas ao seu sistema operacional até que nenhuma saída de versão seja recebida.

Note

Se você usou um gerenciador de pacotes para instalar a AWS CLI (pip, apt, brew etc.), use o mesmo gerenciador de pacotes para desinstalá-la. Siga as instruções fornecidas pelo gerenciador de pacotes sobre como desinstalar todas as versões de um pacote.

[Voltar ao início](#)

O comando AWS CLI processado com um nome de parâmetro incompleto

Possível causa: você usou uma abreviação reconhecida do parâmetro AWS CLI

Como o AWS CLI é construído usando Python, ele AWS CLI usa a argparse biblioteca Python, incluindo o argumento. [allow_abbrev](#) As abreviações dos parâmetros são reconhecidas pelo e processadas AWS CLI .

O exemplo de [create-change-set](#) comando a seguir altera o nome da CloudFormation pilha. O parâmetro `--change-set-n` é reconhecido como uma abreviatura de `--change-set-name` e AWS CLI processa o comando.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

Quando a abreviação pode ser de vários comandos, o parâmetro não será reconhecido como uma abreviação.

O exemplo de [create-change-set](#) comando a seguir altera o nome da CloudFormation pilha. O parâmetro `--change-set-` não é reconhecido como uma abreviação, pois há vários parâmetros dos quais ele pode ser uma abreviação, como `--change-set-name` e `--change-set-type`. Portanto, o AWS CLI não processa o comando.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

Warning

Não use abreviações de parâmetros propositadamente. Elas não são confiáveis e não são compatíveis com versões anteriores. Se algum novo parâmetro for adicionado a um comando que confunda as abreviações, os comandos serão interrompidos. Além disso, se o parâmetro for um argumento de valor único, ele poderá causar um comportamento inesperado com os comandos. Se várias instâncias de um argumento de valor único forem transmitidas, somente a última instância será executada. No exemplo a seguir, o parâmetro `--filters` é um argumento de valor único. Os parâmetros `--filters` e `--filter` são especificados. O parâmetro `--filter` é uma abreviação de `--filters`. Isso faz com que duas instâncias de `--filters` sejam aplicadas e somente o último argumento `--filter` se aplica.

```
$ aws ec2 describe-vpc-peering-connections \  
  --filters Name=tag:TagName,Values=VpcPeeringConnection \  
  --filter Name=status-code,Values=active
```

Confirme se você está usando parâmetros válidos antes de executar um comando para evitar comportamentos inesperados.

[Voltar ao início](#)

Erros de acesso negado

Possível causa: O arquivo do AWS CLI programa não tem permissão de “execução”

No Linux ou no macOS, verifique se o programa `aws` tem permissões de execução para o usuário que está chamando. Normalmente, as permissões são definidas como `755`.

Para adicionar permissão de execução para seu usuário, execute o comando a seguir, substituindo `~/.local/bin/aws` com o caminho para o programa no seu computador.

```
$ chmod +x ~/.local/bin/aws
```

[Voltar ao início](#)

Possível causa: sua IAM identidade não tem permissão para realizar a operação

Exemplo de texto de erro:

```
$ aws s3 ls
An error occurred (AccessDenied) when calling the ListBuckets operation: Access
denied.
```

Quando você executa um AWS CLI comando, AWS as operações são executadas em seu nome, usando credenciais que o associam a uma IAM conta ou função. As políticas anexadas devem conceder a você permissão para chamar as API ações que correspondem aos comandos que você executa com AWS CLI o.

A maioria dos comandos chama uma única ação com um nome que corresponde ao nome do comando. No entanto, comandos personalizados, como `aws s3 sync` chamar vários APIs. Você pode ver qual APIs comando chama usando a `--debug` opção.

Se você tiver certeza de que o usuário ou a função tem as permissões adequadas atribuídas pela política, certifique-se de que seu AWS CLI comando esteja usando as credenciais que você espera. Consulte a [próxima seção sobre credenciais](#) para verificar se as credenciais que ele AWS CLI está usando são as que você espera.

Para obter informações sobre a atribuição de IAM permissões, consulte [Visão geral do gerenciamento de acesso: permissões e políticas](#) no Guia do IAM usuário.

[Voltar ao início](#)

Credenciais inválidas e erros de chave

Exemplo de texto de erro:

```
$ 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.
```

Possível causa: AWS CLI Está lendo credenciais incorretas ou de um local inesperado

AWS CLI Pode estar lendo as credenciais de um local diferente do esperado ou as informações do seu par de chaves estão incorretas. Execute a `aws configure list` para confirmar quais credenciais são usadas.

O exemplo a seguir mostra como verificar as credenciais usadas para o perfil padrão.

```
$ 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

O exemplo a seguir mostra como verificar as credenciais de um perfil nomeado.

```
$ 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

Para confirmar os detalhes do par de chaves, verifique os arquivos `config` e `credentials`. Para obter mais informações sobre os arquivos `config` e `credentials`, consulte [the section called “Configurações do arquivo de configuração e credenciais”](#). Para obter mais informações sobre credenciais e autenticação, incluindo a precedência de credenciais, consulte [Autenticação e credenciais de acesso](#).

[Voltar ao início](#)

Causa possível: o relógio do computador está fora de sincronia.

Se você estiver usando credenciais válidas, seu relógio poderá estar fora de sincronia. No Linux ou macOS, execute `date` para verificar a hora.

```
$ date
```

Se o relógio do sistema não estiver correto dentro de alguns minutos, use `ntpd` para sincronizar.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

No Windows, use as opções de data e hora no Painel de controle para configurar o relógio do sistema.

[Voltar ao início](#)

Assinatura não corresponde aos erros

Exemplo de texto de erro:

```
$ 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 executa um comando, ele envia uma solicitação criptografada aos AWS servidores para realizar as operações de AWS serviço apropriadas. Suas credenciais (a chave de acesso e a chave secreta) estão envolvidas na criptografia e permitem AWS autenticar a pessoa que fez a solicitação. Há vários fatores que podem interferir na operação correta desse processo, conforme indicado a seguir.

Possível causa: Seu relógio está fora de sincronia com os AWS servidores

Para ajudar a proteger contra [ataques de reprodução](#), a hora atual pode ser usada durante o processo de criptografia/descriptografia. Se a hora do cliente e a hora do servidor não coincidirem além do permitido, o processo poderá falhar e a solicitação será rejeitada. Isso também pode

acontecer quando você executa um comando em uma máquina virtual cujo relógio está fora de sincronia com o relógio da máquina host. Uma causa possível é quando a máquina virtual hiberna e, depois de ativada, demora algum tempo para sincronizar o relógio com a máquina host.

No Linux ou macOS, execute `date` para verificar a hora.

```
$ date
```

Se o relógio do sistema não estiver correto dentro de alguns minutos, use `ntpd` para sincronizar.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

No Windows, use as opções de data e hora no Painel de controle para configurar o relógio do sistema.

[Voltar ao início](#)

Possível causa: seu sistema operacional está manipulando incorretamente AWS as chaves que contêm determinados caracteres especiais

Se suas AWS chaves incluírem determinados caracteres especiais, como `-`, `,` ou `+ /%`, algumas variantes do sistema operacional processarão a cadeia de caracteres de forma inadequada e farão com que ela seja interpretada incorretamente.

Se você processar suas chaves usando outras ferramentas ou scripts, como ferramentas que criam o arquivo de credenciais em uma nova instância como parte de sua criação, essas ferramentas e scripts podem ter seu próprio tratamento de caracteres especiais, o que faz com que sejam transformados em algo que AWS não reconhece mais.

Sugerimos gerar novamente a chave secreta para obter uma que não inclua o caractere especial que está causando problemas.

[Voltar ao início](#)

Erros de nenhum console do Windows encontrado

Exemplo de texto de erro:

```
$ aws s3 ls
No Windows console found. Are you running cmd.exe?
```

Ao usar um AWS CLI comando, você recebe a mensagem “Nenhum console do Windows encontrado”. Are you running cmd.exe?” (Nenhum console do Windows encontrado. Você está executando o cmd.exe?”. Isso geralmente é um erro visto na AWS CLI versão 1 se o Python `prompt_toolkit` que você instalou estiver desatualizado. Para resolver esse problema, instale uma versão recente do `prompt_toolkit` no [site do Python](#).

[Voltar ao início](#)

SSLerros de certificado

Possível causa: AWS CLI Não confia no certificado do seu proxy

Exemplo de texto de erro:

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Ao usar um AWS CLI comando, você recebe uma mensagem `[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed` de erro. Isso é causado pela AWS CLI falta de confiança no certificado do proxy devido a fatores como o certificado do proxy ser autoassinado, com sua empresa definida como Autoridade de Certificação (CA). Isso impede que eles encontrem o certificado raiz AWS CLI da CA de sua empresa no registro local da CA.

Para corrigir isso, instrua o arquivo de AWS CLI onde encontrar suas empresas usando a [ca_bundle](#) configuração do `.pem` arquivo de configuração, a opção de linha de [--ca-bundle](#) comando ou a variável de [AWS_CA_BUNDLE](#) ambiente.

[Voltar ao início](#)

Causa possível: sua configuração não está apontando para o local correto do certificado raiz da CA

Exemplo de texto de erro:

```
$ aws s3 ls
SSL validation failed for regionname [Errno 2] No such file or directory
```

Isso é causado devido ao local do arquivo do pacote da Autoridade de Certificação (CA) estar configurado incorretamente na AWS CLI. Para corrigir isso, confirme o local em que o arquivo `.pem` está localizado e atualize a configuração da AWS CLI usando o arquivo de configuração [ca_bundle](#), a opção de linha de comando da `--ca-bundle` ou a variável de ambiente [AWS_CA_BUNDLE](#).

[Voltar ao início](#)

Possível causa: sua configuração não está usando o correto Região da AWS

Exemplo de texto de erro:

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Erros ou resultados inesperados podem ocorrer se um AWS service (Serviço da AWS) não estiver disponível para o que você especificou Região da AWS ou se seus recursos estiverem localizados em um local diferente Região da AWS. Para obter etapas sobre a solução de problemas, consulte [the section called “Verifique o que Região da AWS seu AWS CLI comando está usando”](#).

[Voltar ao início](#)

Possível causa: sua TLS versão precisa ser atualizada

Exemplo de texto de erro:

```
$ aws s3 ls
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

AWS service (Serviço da AWS) Está usando uma versão TLS que é incompatível com a TLS versão do seu dispositivo. Para resolver esse problema, atualize para uma TLS versão compatível. Para ter mais informações, consulte [the section called “Aplicando uma versão mínima TLS”](#).

[Voltar ao início](#)

Erros inválidos JSON

Exemplo de texto de erro:

```
$ 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}
```

Ao usar um AWS CLI comando, você recebe uma mensagem de erro `Invalid JSON`. Isso geralmente é um erro visto quando você insere um comando com um JSON formato esperado e AWS CLI não consegue lê-lo JSON corretamente.

Possível causa: Você não inseriu um JSON valor válido AWS CLI para uso do

Confirme se você tem JSON uma entrada válida para seu comando. Sugerimos usar um JSON validador JSON se você estiver tendo problemas de formatação.

Para um JSON uso mais avançado na linha de comando, considere usar um JSON processador de linha de comando, por exemplo `jq`, para criar JSON cadeias de caracteres. Para obter mais informações sobre `jq`, consulte o [repositório jq](#) em GitHub

[Voltar ao início](#)

Possível causa: as regras de cotação do seu terminal estão impedindo o envio de documentos válidos JSON para o AWS CLI

Antes de AWS CLI receber qualquer coisa de um comando, seu terminal processa o comando usando suas próprias regras de citação e escape. Devido às regras de formatação de um terminal, parte do seu JSON conteúdo pode ser removida antes que o comando seja passado para o AWS CLI. Ao formular comandos, use as [regras de aspas do seu terminal](#).

Para solucionar problemas, use o `echo` para ver como o shell está lidando com seus parâmetros:

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}  
ReadCapacityUnits:15 WriteCapacityUnits:10
```

```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'  
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

Modifique seu comando até que seu até que JSON seja válido seja retornado.

Para uma solução de problemas mais detalhada, use o parâmetro `--debug` para visualizar os logs de depuração, pois eles exibirão exatamente o que foi passado para 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']
```

Use as regras de cotação do seu terminal para corrigir quaisquer problemas que sua JSON entrada tenha ao ser enviada para o AWS CLI. Para obter mais informações sobre regras de aspas, consulte [the section called “Aspas com strings”](#).

Note

Se você estiver tendo problemas JSON para obter a validade do AWS CLI, recomendamos ignorar as regras de cotação de um terminal para entrada de JSON dados usando Blobs para passar seus JSON dados diretamente para o AWS CLI. Para obter mais informações sobre o Blobs, consulte [Blob](#).

[Voltar ao início](#)

Recursos adicionais

Para obter ajuda adicional com seus AWS CLI problemas, visite a [AWS CLI comunidade](#) em GitHub ou a [AWS re:Post comunidade](#).

[Voltar ao início](#)

Histórico do documento do guia do usuário da AWS CLI

A tabela a seguir descreve adições importantes feitas ao Manual do usuário da AWS Command Line Interface a partir de janeiro de 2019. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em o feed RSS.

Alteração	Descrição	Data
Informações atualizadas sobre credenciais e autenticação.	Instruções e exemplos atualizados de credenciais e métodos de autenticação. Isso inclui a atualização de páginas de configuração relevantes. Para acomodar esse aumento na documentação, tópicos relevantes sobre credenciais foram movidos para a nova seção Autenticação e credenciais de acesso .	31 de março de 2023
Os conteúdos referentes à AWS CLI V1 e V2 agora estão separados nos respectivos guias	Para maior clareza e facilidade, o conteúdo da AWS CLI versão 1 e da AWS CLI versão 2 agora está separado nos próprios guias. Para a AWS CLI versão 2, consulte o Guia do usuário da AWS Command Line Interface mais recente	2 de novembro de 2021
Adição de informações sobre o alias AWS CLI.	Adição de informações sobre o alias AWS CLI. Os aliases são atalhos que você pode criar na AWS Command Line Interface (AWS CLI) para encurtar comandos ou scripts que utiliza com frequência.	11 de março de 2021

Informações atualizadas sobre saídas de filtros	Informações sobre filtros atualizadas e movidas para suas próprias páginas.	1º de fevereiro de 2021
Anúncio de descontinuação do Python 2.7, 3.4 e 3.5	O Python 2.7 foi descontinuado pela Python Software Foundation em 1º de janeiro de 2020. No futuro, os clientes que usam a AWS CLI versão 1 deverão fazer a transição para o Python 3 (no mínimo, Python 3.6). O suporte ao Python 2.7 foi descontinuado para novas versões da AWS CLI versão 1 a partir de 19/7/2021 . As versões Python 3.4 e 3.5 foram descontinuadas a partir de 1/2/2021.	29 de janeiro de 2021
Adição de exemplo de script do Amazon S3	Adição de um exemplo de script de ciclo de vida do Amazon S3.	15 de outubro de 2020
Adição de exemplo de script do Amazon EC2	Adição de um exemplo de script de tipo de instância do Amazon EC2.	15 de outubro de 2020
Adição de informações sobre novas tentativas	Adição de uma página sobre novas tentativas para recursos e comportamento de novas tentativas na AWS CLI.	17 de setembro de 2020
Página sobre paginação nos lados do servidor e do cliente	Informações sobre paginação atualizadas e centralizadas em uma única página.	17 de agosto de 2020

Página de comandos do s3 atualizada	Página de comandos de alto nível do s3 atualizada com novos exemplos e recursos.	30 de julho de 2020
Atualização de informações sobre a instalação	As informações de instalação, atualização e desinstalação para Linux, macOS e Windows foram atualizadas.	19 de maio de 2020
Atualizado para remover o suporte ao Python 2.6 e 3.3 da AWS CLI versão 1	Desde de 10 de janeiro de 2020, a AWS CLI versão 1 não oferece mais suporte ao uso das versões 2.6 ou 3.3 do Python. Atualize para uma versão mais recente do Python para usar a AWS CLI versão 1.17 ou posterior.	10 de janeiro de 2020
Nova seção MFA	Adicionada uma nova seção que descreve como acessar a CLI usando a autenticação multifator e funções.	3 de maio de 2019
Atualização da seção "Uso da CLI"	Principais melhorias e adições às instruções e aos procedimentos de uso.	7 de março de 2019
Atualização da seção "Instalar a CLI"	Principais melhorias e adições às instruções e aos procedimentos de instalação da AWS CLI.	7 de março de 2019
Atualização da seção "Configurar a CLI"	Principais melhorias e adições às instruções e aos procedimentos de configuração da AWS CLI.	7 de março de 2019